AN OBJECT-ORIENTED APPROACH TO STRUCTURING MULTICRITERIA DECISION SUPPORT IN NATURAL RESOURCE MANAGEMENT PROBLEMS

By

DINGFEI LIU

Thesis Presented for the Degree of

DOCTOR OF PHILOSOPHY

Under the Supervision of Professor Theodor J Stewart

Department of Statistical Sciences

UNIVERSITY OF CAPE TOWN

August 2001

Declaration

I, Dingfei Liu, hereby declare that this thesis is my own work and that it has not been submitted for a degree at another university.

Signed by candidate

August, 2001

To my parents

ACKNOWLEDGEMENTS

I am indebted to the following people who have offered valuable assistance and support during the preparation of this thesis.

My supervisor, Professor Theodor J Stewart, without whom this thesis would not have been completed, for his erudite, enthusiastic and patient guidance throughout the entire duration of my studies, and for his careful examination of every line of this thesis before the submission. I really enjoy working with you, and appreciate you very much for being always there when needed.

Ms Alison Joubert, for her sustained interest in the topic, her invaluable assistance at the workshops, and her insightful comments.

Professor Gregory Kersten at Concordia University, Professor Simon French at the University of Manchester, Professor Räimo Hämäläinen at the Helsinki University of Technology, Professor Jonathan Rosenhead at the London School of Economics, Professor Shouhong Wang at the University of Massachusetts Dartmouth, Professor Dave Bustard at the University of Ulster, and Professor Tung Bui at the University of Hawaii, for their help and useful suggestions at various stages during the preparation of this thesis.

My fellow colleagues in the Department of Statistical Sciences at the University of Cape Town, who showed constant interest in my work.

The University of Cape Town, for the provision of the scholarship towards the research.

My wife, Yue Lu, for her long patience and support.

My parents, my sisters and my brother who encouraged me all the way through.

ABSTRACT

The undertaking of MCDM (Multicriteria Decision Making) and the development of DSSs (Decision Support Systems) tend to be complex and inefficient, leading to low productivity in decision analysis and DSSs. Towards this end, this study has developed an approach based on object orientation for MCDM and DSS modelling, with the emphasis on natural resource management.

The object-oriented approach provides a philosophy to model decision analysis and DSSs in a uniform way, as shown by the diagrams presented in this study. The solving of natural resource management decision problems, the MCDM decision making procedure and decision making activities are modelled in an object-oriented way. The macro decision analysis system, its DSS, the decision problem, the decision context, and the entities in the decision making procedure are represented as "objects". The object-oriented representation of decision analysis also constitutes the basis for the analysis of DSSs.

Classes of decision elements and primary DSS components are identified as a result of the analysis of the generic system requirements for DSSs and the consideration of DSS evaluation principles. These classes form the basis of the definition of the problem, a kind of DSS building material, and a fundamental type of knowledge for decision making and DSS development. Classes and their interactions can represent the activities of decision making as well as the major functions of DSSs.

A MCDM framework including four processes, i.e., initial understanding (IU), strategic analysis - brainstorming - decision element identification (SBI), structuring, and exploring, and methodological guidelines are proposed for object-oriented MCDM. Basic ways and general processes to utilise object orientation in MCDM are illustrated. The framework is able to facilitate decision making processes of problem identification, problem analysis and structuring, evaluation, choice and implementation.

Finally, practical implementations of the approach are demonstrated. The DSS model, which mainly consists of the general DSS requirements, classes, and design of subsystems and system architecture, is applied to the development of a DSS for water resources allocation. It is shown that the DSS model is very useful in the DSS development. The implementation of the object-oriented MCDM is demonstrated by involving groups of students representing non-expert public in the controversial problem of alien vegetation removal on Table Mountain in South Africa. It is shown that the decision facilitation method can produce quick problem understanding and easy problem analysis and formulation.

It is concluded that the object-oriented approach can provide effectiveness and efficiency in both the development of DSSs and facilitation of decision analysis. The approach provides a solid methodological and philosophical basis for both decision making and the modelling of DSSs. Several directions for future research are identified.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
DBMS	Data Base Management Systems
DSS	Decision Support System
ES	Expert System
GDSS	Group Decision Support System
LAN	Local Area Network
MIS	Management Information Systems
MS/OR	Management Science/Operations Research
MCDA	Multicriteria Decision Aids
MCDM	Multicriteria Decision Making
SSM	Soft Systems Methodology
SODA	Strategic Options Development and Analysis

Chapter 1

Introduction

1.1 Prologue

Humans make decisions all the time. In fact, our ability to make decisions makes us human. Most decisions, such as stopping the car before a red traffic light, are made naturally. Some decisions, such as buying a car, need a greater degree of deliberation. Some decisions, such as doing business in car dealing, need some professional skills. There are still some decisions, such as allocating natural resources like land and water, which cannot even be conceived without a comprehensive analysis of alternatives and consequences.

There is clearly a need for the study of decision making, a concept which is usually called "decision analysis" in the literature (Keeney, 1982; von Winterfeldt and Edwards, 1986; French, 1998). The study would facilitate decision making activities, especially problem analysis and problem structuring, which lead to the understanding and the identification of the problem. Multicriteria Decision Making (MCDM), which is the study of decision making for problems with multiple objectives, has been applied (Stewart, 1992; Dyer, Fishburn, Steuer and Wallenius, 1992; Gal, Stewart and Hanne, 1999) in solving complicated decision problems such as natural resource management, which has received much attention in research.

Natural resource management problems tend to have an extensive coverage and far-reaching effects and repercussions. Examples are problems related to water, forestry, land, etc. The effects of these problems on health and nature are accompanied by social, political, and economic functions, locally, internationally, and even globally. The number of people affected in natural resource management and thus the number of participants in the decision making processes are very large in complex cases. The increased awareness of the seriousness of natural resource management has led to a considerable boom in the research of decision making in this area, and especially, has given rise to a need for effective and efficient computer-aided decision support.

The role of the computer in decision making is especially influential due to the limited information processing capability of human memory, computational skills and attention span. The potential benefits of computer-aided support for decision making can be divided into two main categories: displaced cost and added value (Carlson, 1983). Displaced cost results from the reduction of costs related to data gathering, computation and data presentation in support of decision making. In these mechanical tasks, the financial value of computer support is measurable and highly valuable. In comparison with non-computer-supported decision making, the added value of computer support can be attributed to obtaining more decision alternatives, to effecting more sophisticated analyses and evaluation of alternatives, to speedier decision making, etc. In the end, even small improvements in decision making can ultimately result in high added value. An integrated human-computer information processor concept (Jacob, Moore and Whinston, 1989) allows humans to interact directly with the environment, while the computer stores historical and problem solving information. In advanced decision making systems, computers can analyse data directly from the environment and/or are able to help humans to make decisions. Given the inherent capabilities and skills of both humans and computers, such an integrated system will certainly complement each other to solve decision making problems more effectively and efficiently.

A computerised decision making system is called a "Decision Support System (DSS)". The precise definition of DSS, however, varies in the literature (Gorry and Scott Morton, 1971; Sprague and Watson, 1986). A widely accepted definition is that found in Gorry and Scott Morton (1971), namely "interactive computer based systems that help decision makers utilise data and models to solve unstructured problems". Janssen (1991), however, has argued that as a result of the absence of a theoretical basis of DSSs, DSSs are developed as instruments for decision support rather than as the ultimate result of a firm theory.

Efficient and philosophically sound methods are needed for decision analysis and DSS development. This is because of the complexity of decision analysis, especially in problem analysis and structuring, and in the development of DSSs for natural resource management. Problem analysis and structuring are the initial and most critical phases in decision analysis according to Adams, Courtney and Kasper (1990).

In this study, a methodology for decision analysis and DSS modelling for MCDM in natural resource management is developed. This methodology is based on an object-oriented philosophy, which holds the view that the world and its constituent parts are composed of independent yet interactive physical or non-physical objects. The DSS model developed on the basis of this methodology is further strengthened by the integration of DSS evaluation principles. This ensures that the performance of a specific DSS implemented will meet the general and most needed requirements for such a system.

The objective of this chapter is to introduce the various concepts used in this study. These concepts include those of decision analysis, MCDM, natural resource management, object orientation, DSSs, and DSS modelling. This chapter also provides basic ideas about the aim, the scope and the limitations of the present study.

This chapter is organised as follows. Decision analysis and MCDM are introduced in Sections 1.2 and 1.3. Sections 1.4 and 1.5 discuss natural resource management, which is the domain of the decision problems to be concentrated on in the study, and its features. The concepts of DSSs are defined in Section 1.6. Section 1.7 introduces the concepts of object orientation, specifically of object-oriented modelling, and presents the advantages of object orientation. Section 1.8 discusses the modelling of decision making and DSSs, and some characteristics of these systems. Finally, the objectives, the scope and the limitations of the study are included in the final section of this chapter.

1.2 Decision Analysis

Decision analysis is a combination of "philosophy, methodology, practice and application useful in the formal introduction of logic and preferences" (Howard, 1968) to decision making in the world. French (1998) tentatively suggests the definition of decision analysis as the study of models (not necessary mathematical) that represent the judgements of decision participants involved in decision making and can lead to a "usually but not always rational" deliberate choice. The research in decision analysis has been very active over the past four decades. Decision analysis has been very successful in practice. Applications of decision analysis have been found in many areas, including energy, manufacturing and services, medical, public policy, etc (Corner and Kirkwood, 1991).

Decision analysis captures the important elements of a decision problem. Decisions inevitably involve people. People participating in a decision are called decision participants, who may be categorised as decision makers (DM), analysts and domain experts. A decision maker is a person who is intrinsically involved and must have at least a sense of, if not full, ownership of the decision problem, although this does not necessarily mean that this individual knows precisely what the problem is. For example, the owner of a car is the decision maker when this person wants to sell the car. Another example of decision maker is a hydroelectric company that is to manage a dam to be built over a river for electricity supply, and has some measure of responsibility for the decision making concerned with the construction of the dam. A decision analyst is a person who is able to use some expertise, especially decision making skills, to help other people involved in the decision to discuss, explore, formulate and analyse the problem in order to make a better informed decision. Domain experts are the people who can offer relevant information and knowledge in some specific areas related to the decision problem. The simplest decisions involve only one decision maker, who provides all the knowledge necessary, makes judgements, performs the analysis and makes a decision. Other decisions involve a group of decision makers as well as analysts and domain experts.

Other main elements of a decision problem include decision alternatives, objectives, criteria, and a choice. A decision alternative is a course of action towards the solution of the decision problem. In the case of car purchase, for example, there are many options such as buying a new Ford, a second-hand Ford, a new Opel, a second-hand Opel, etc. An objective represents what decision makers wish to achieve in solving the decision problem. Criteria represent particular points of view or interests according to which decision alternatives may be compared. An objective is a quantified representation of a criterion, usually expressed in the form of a desire to maximise or to minimise a measurable property of the decision problem. For instance, economy in fuel consumption may be one of the criteria in car purchase while to minimise the fuel cost is an objective of the problem. A choice is the final selection from a pool of decision alternatives by applying some techniques of decision making, for example, the final selection of the car to buy.

Decision analysis usually begins when there is a need to solve the decision problem. The decision makers may be aware of some issues of the problem, but they might have no clear idea of what the choices before them are, and even less about how to evaluate those choices. Thus, they seek help from a decision analyst. The decision analyst may help the decision makers to identify and formulate the decision problem by using some decision making skills. After the formulation of the decision problem, the analysis elicits value judgements and other relevant information about the decision alternatives from the decision makers. In both formulating the problem and eliciting values, the decision analyst may consult some domain experts for information with regard to specific issues. Necessary data is then collected for calculations according to decision models, which are the computational forms of decision analysis methods. The results of the calculations are presented back to the decision makers.

Decision analysis offers techniques for analysing, structuring and solving single or multiple criteria decision problems. The decision making problems which we confront in the real world, however, are mainly those of multicriteria.

1.3 Multicriteria Decision Making (MCDM)

Multicriteria Decision Making (MCDM) is a general term referring to the collection of procedures and methods used for solving decision problems in contexts in which conflict between goals or criteria exists (Stewart, Scott and Iloni, 1993). Some other terms, such as Multicriteria Decision Analysis (MCDA), Multiattribute Decision making (MADM), and Multiobjective Decision making (MODM), are often used to emphasise specific aspects of the problem or approach. As with general decision analysis, MCDM captures the essential elements of decision problems involving more than one criterion. They include decision alternatives, objectives, criteria, choice, and various kinds of decision participants, such as decision makers, decision analysts and domain experts. Virtually all decisions involve multiple criteria, and fall into the domain of MCDM.

The major tasks of multicriteria decision analysis include problem analysis, problem structuring, decision alternative evaluation, criterion comparison, and choice. Problem analysis results in the understanding and the identification of a decision problem and its elements. Problem structuring generates decision alternatives and criterion structures. Decision alternative evaluation enables decision makers to distinguish preferentially between the alternatives under consideration. Criterion comparison, as the name indicates, is about finding out the relationships among the criteria and representing their relative importance. Criterion comparison is usually carried out on the basis of criterion structures. Choice means that the decision is made based on the evaluation of alternatives and criterion comparison. These tasks may be iterative, as new findings about the problem are located in the procedure of decision making.

Over the past three decades, MCDM has experienced a striking development period both in practice and theory, and has developed into a discipline in its own right (Stewart, 1992; Dyer, Fishburn, Steuer and Wallenius, 1992). In the 1970s, MCDM research focused on the theoretical foundation of multiple objective mathematical programming. Examples are Fishburn (1970), Benayoun, Montgolfier, Tergny and Larichev (1971), Dyer and Feinberg (1972), Lee (1972), Roy (1973), Zeleny (1975), Ignizio (1976), Keeney and Raiffa (1976), Zionts and Wallenius (1976), Steuer(1977), Cohon (1978), and Benayoun, Montgolfier, Tergny and Bitran (1979). Since the 1980s, emphasis has shifted towards the implementation of computerised DSSs. A DSS was first characterised as "a computer-based information system" to facilitate decision making activities (Ginzberg and Stohr, 1982). Some multiple criteria DSSs (Jelassi, Jarke and Stohr, 1985; Belton and Vicker, 1989) have been implemented to improve decision making in organisations. At the same time, much attention has been paid to the practical applications of MCDM in a wide range of areas, especially in natural resource management (Hallefjord and Jornsten, 1986; Talcott, 1992; Ellis, 1992; Stewart, Scott and Iloni, 1993; Stewart, Joubert, Scott and Low, 1996).

1.4 Natural Resource Management

Natural Resources, either stock (non-renewable, such as all minerals and land) or flow (renewable, including water, air, animal, plant life, solar radiation, wind power and tidal energy), have played a critical role in the life of humans. Over the last several decades, a plethora of highly diverse natural resource crises has attracted public, political and academic attention. These include concerns over declining environmental quality, the irreparable damage inflicted on global ecological systems, the suffering caused by drought and desertification, the imminent physical exhaustion of mineral stocks, the economic and security threats posed by politically motivated disruptions of the resource trade, and the possible collapse of the world economical system.

It is possible to divide the extremely heterogeneous range of influences of natural resources into two not necessarily distinct phases (Rees, 1985). The first phase focuses largely on the physical environment, the limits of the resources and their deteriorating quality. The second phase involves a re-identification of the critical resource problems and shifts attention from physical scarcity and environmental deterioration to a broader investigation of the social, economic and political dimensions of resource utilisation resulting from natural resource management.

Natural resource management deals with decision making in the distribution of natural resources, and the wealth or welfare derived from them over space and time. It is a real life problem, which is closely related to the well being of people, such as employment and health. It might contain multiple concerns to the society. The natural world is made up of complex ecosystems, many of which depend on quite delicate balances. If the society upsets the balance, by pouring toxic chemicals into the atmosphere, for example, the effect is felt downwind where the acid rain kills trees, poisons lakes and wipes out other productive activity (Edwards, 1995).

In the social dimension, the greatest challenge to natural resource management is the steady increase in population and its demand for more natural resources. The poor development and mismanagement of natural resources might be the potential source of revolutions, political instability, conflicts, famines, pandemic diseases, and poor economic growth. In natural resources' management projects, economic merit is generally the dominant criterion assessed. But, other intangible and indirect goals, such as the interest displayed by the public in recreation, the quality of the environment and the aesthetic possibilities presented, are also important. These intangible socio-political criteria, which are not immediately expressible in numerical terms, need to receive their appropriate attention from decision makers.

However, the ultimate decision making problem in natural resource management has been the lack of consensus amongst the different groups or individuals who have conflicting social, economic and political interests, and who benefit from the exploitation of the natural resources. Natural resource management is actually a MCDM decision problem. It is also a complex societal problem, a term defined by DeTombe (1994, 1996) to address real life problems, due to its dynamic character, the number of people involved, and the impact it has on society. It in fact may include local concerns in and between organisations, issues of national governments, and global and international aspects. Detailed features of natural resource management are described in the next section.

1.5 Features of Natural Resource Management

As discussed above, natural resource management is complex due to its multiple dynamic aspects. Conditions of natural resources change over time. People who have a role to play in the allocation have conflicting objectives. The major features of natural resource management are summarised as follows:

(1) Long Term Planning

Natural resource management inevitably falls into the category of long term planning due to the fact that most critical natural resources, such as all minerals and land, are non-renewable, or if renewable, such as water and air, they have a lasting impact on the environmental quality or on the society. Policy planning and development of relatively long-term planning of resources are done in a context of very dramatic changes. Value judgements about the decision are divergent among various decision participants. There are difficulties in assessing the consequences of decision alternatives, which have long-term influences on different issues concerned.

(2) Uncertainty in Information

Because of its complexity and its long-term characteristics, natural resource management contains various types of uncertainty, such as long term environmental consequences, future political, social and economic conditions, and the dynamic relationships between them. These uncertainties need to be taken into consideration when decisions about the allocation of natural resources have to be made.

(3) Measurement of Goal Achievement

In the decision making of natural resource management, it becomes problematic when trying to measure and compare the goals of decisions. It is clear, from the literature survey by Stewart, Scott and Iloni (1993), that the way that the achievement of any specific goal is expressed is one of two main issues in goal measurement. Some unambiguous and tangible goals, for example, minimising operating costs of a dam, are easy to calibrate. Goals expressed in economic terms generally fall into this category. In natural resource management, it is problematic to express all the objectives in economic terms since some criteria at least are highly politically charged. Other goals, which can be best described as less tangible, or even intangible, such as the maintenance of environmental quality, are difficult to measure.

The other issue in goal measurement is the comparison of goals which are not directly commensurate. Different units, e.g. levels of pollution and of unemployment rate, are usually used to express goals, but even those expressed in obviously similar units, for example, the fixed and operating costs of a project, may not truly be commensurate as they may be funded from very different budgets.

The ultimate objective in the goal measurement for natural resource management is to achieve some proper means to calibrate both tangible and intangible goals in such a way that they are commensurate to a certain degree so that levels of achievement of different goals can be related.

(4) Conflicting Objectives

There are likely to be fairly high levels of conflicts in decision making regarding natural resource management. Different groups or individuals involved in natural resource management have conflicting social, economic and political interests. Conflicts among various objectives are usually dramatic and substantial. They have to be resolved somehow, and consensus amongst these people has to be sought to make a decision.

(5) Number of Decision Alternatives

In natural resource management, the number of decision alternatives identified may be extremely large. This is mainly because of the complexity and the interaction of multiple aspects involved in the decision making of natural resource management. Several decision alternative elements, which are the instruments or components of a decision alternative, may be identified together with the ranges of freedom of action, usually in the form of a grid of values, for each element. Even with relatively coarse grids of values, however, the size of a decision alternative set may be too large to be practical to work with. Generally, hundreds of possible decision alternatives of combination could be produced even though only five or six decision alternative elements and three or four options for each element exist. The pool of decision alternatives identified may nevertheless still not be sufficiently rich for all interested parties to find a reasonably satisfactory alternative. Thus, a set of decision alternatives needs to be manageable in size so that decision participants can directly evaluate the alternatives in the set, and it also needs to be representative enough to satisfy all interested parties.

These are the main features of natural resource management. It seems that decision making in this area may be very difficult because of its complexity. Various methods, especially MCDM, have been applied in solving these decision problems. The resulting multicriteria DSSs, generally computer based, are developed to provide decision support.

1.6 Decision Support Systems (DSSs)

To effectively support some or all the phases of decision making in solving semi-structured or unstructured decision problems such as natural resource management problems, an interactive computer system is very important and useful. This kind of computer system is called a Decision Support System (DSS), which was coined in the 1970s (Gorry and Scott Morton, 1971; Gerrity, 1971; Keen and Scott Morton, 1978) based on the concepts of electronic data processing (EDP) and management information systems (MIS).

Since the late 1970s, DSS researchers, aided by rapid advances in information technology, have sought ways to improve automated support for decision making in organisations. There are three major areas of research in decision support tools (Sprague, 1980, Nunamaker, Applegate and Konsynski, 1988). The first involves the development of the tools that decrease the cost and time necessary to implement and maintain single model and single problem decision support algorithms. Examples of these tools include spreadsheets, e.g., MicroSoft EXCEL, LOTUS 1-2-3 and SUPERCALC, modelling languages, e.g., GPLAN, and general system design tools, e.g., graphics packages and fourth generation languages. Coupled with the rapid increase in the use of microcomputers and personal workstations, these tools have led to the widespread use of decision support algorithms within organisations. The second area involves the development of specific DSSs that enable the application of DSSs for a specific problem. These kinds of systems usually contain more comprehensive functions than DSS tools. They support many stages of decision making activities. A third major area of DSS research involves the development of generalised DSS generators that allow the support of multiple problems using a variety of data sources and models, and also to centralise management of organisational models.

DSSs include various subcategories from different points of view. From the application point of view, DSSs may include specific DSSs, DSS generators and DSS tools as suggested by Sprague (1980). According to Sprague (1980), a DSS generator is a "package of related hardware and software which provides a set of capabilities to build specific DSSs quickly and easily". A detailed explanation is given by Bhargava, Sridhar and Herrick (1999) to the effect that a DSS generator only provides certain generic functions and visual interfaces to model and analyse different specific decision problems under the same application environment. A DSS generator is akin to an interpreted programming language environment – you must always have the interpreter to execute the code. Specific DSSs are systems that actually support the solution of specific sets of related decision problems while DSS tools are the hardware and software elements to facilitate the development of both specific DSSs and DSS generators. Specific DSSs can be developed either directly from tools or by adapting the DSS generator to meet client's requirements.

From the academic research point of view, however, DSSs may include three main categories, including group DSSs, intelligent DSSs and distributed DSSs. Group Decision Support Systems (GDSS) support decision making through telecommunication and networks to groups consisting of individuals in different places and at different times. Intelligent Decision Support Systems (IDSS) result from the interdisciplinary combination of artificial intelligence, particularly expert systems and knowledge engineering, and the traditional DSS methods. Distributed Decision Support Systems (DDSS) encompass many physically separated but logically related information processing nodes, each of which contains some facilities capable of decision support.

DSSs for natural resource management are usually GDSS. Kreamer and King (1988) define GDSS as interactive computer-based systems that facilitate the solution of ill-structured problems by a group of decision makers working together as a team. In the case of natural resource management, the "decision makers" are actually the governmental organisations or their entrusted agents, who may make the final decision about the resource allocation. Stakeholders in fact make use of a GDSS to solve the problems as a group. The group members share information interactively. This results in improved effectiveness of the decision making. The group decision making goals may include generating ideas and actions, choosing alternatives and negotiating solutions (McGrath, 1984).

Generally, the term DSS is used to mean DSS generators unless it is otherwise indicated. Specific DSSs might be unaffordable to implement and specific DSSs can only apply to a specific decision problem. Examples of specific DSSs can be found among the systems developed specifically for certain clients. These specific DSSs normally need enormous investment of time and capital. On the other hand, the DSS generator, which is also known as the DSS shell, can be used in different decision situations and can also be used to create a subsystem for a specific decision problem. There are some commercial examples such as VISA, Decision Explorer (formally called Graphics Cope) and ELECTRE. Besides commercial packages, DSS generators for some specific problem domains, such as health, finance, transportation and natural resources, are especially desirable and useful to the practitioners of decision analysis in these fields. DSS development therefore generally means the procedure and the resources used to create a DSS generator, either commercial or non-commercial. One interesting phenomenon in DSS development is that many DSSs are implemented with object-oriented programming languages based on object orientation, which is covered in the next section.

1.7 Object Orientation

The methodology of object orientation originated from object-oriented programming, which in turn originated in simulation modelling aided by the development of the discrete event simulation language, Simula, in the late 1960s. Interest shifted to object-oriented design after object-oriented programming began to mature. Object-oriented analysis became a major area of attention in the late 1980s.

1.7.1 Basic Concepts of Object Orientation

According to object orientation, the real essence of the world and its systems relates to their concepts rather than their functionality. The functionality is important but it is not of overriding importance. The concepts of the world and the components of it have changed least over time. For example, there is a strong resemblance between a conceptual diagram of a computer system constructed in the 1960s and that of the computer system 30 years later. Another example is ourselves – the human beings. Over thousands of years, the concept of human being is still the same even though what modern human beings are capable of doing has become much more sophisticated. Object-oriented approaches help to organise the functions around the concept.

The common element in object orientation is the "object". In object-oriented systems, all real world entities, either concrete or abstract, are treated as objects. The most general version of the object paradigm (Selic, Gullekson and Ward,1994) defines an object as a logic machine of whatever level of granularity, which may be interconnected with other logic machines to construct a system. The objects in a system may intercommunicate by receiving and sending messages to each other.

Objects in the world are categorised as a hierarchy. A set of object instances of the same nature is collectively organised as a class. Objects can be represented as (Sully 1993):

Class (name of the class, for example Human being class)
+instance (name) (the particular occurrence of the class, for example "Jones")
+services (operations) (for example "Sleep")
+arguments (attributes) (for example sex, date of birth)

Each object has a state, which is expressed by a set of values and the status of the operations. Objects encapsulate different properties of the entities, such as arguments (attributes) and services (operations or behaviours). The attributes and operations encapsulated in an object can only be accessed by passing messages to that object. Encapsulation is one of the fundamental ideas behind the object-oriented approach. Objects with the same or similar attributes are grouped as a "class". The instances of a class are objects which are so similar that they share the same set of attributes. Related messages are usually designed as standardised sets called "protocols".

Object Orientation views the world as made up of objects and messages. An object is a complete entity which performs a definite task and contains all components needed to carry out the task. The link between an object and its environment, other objects, is made via messages. The effect of a message on an object depends both on the message and the receiving object, that is, the same message may cause different actions from two different objects. The actions of individual entities and those of the community or of the world as a whole are invoked by the "messages" among the objects. Objects communicate by passing messages. Objects respond to messages by selecting a corresponding method to execute the message received. "Methods" are also called "operations" or "behaviours" in the objects.

Besides encapsulation, another major feature of object orientation is "inheritance". "Subclasses" can inherit the properties of their "superclasses" and can also have some special properties of their own. Specialisation of an object class would automatically inherit the general characteristics of the class. Child classes may inherit some features from the parent classes while being able to override other attributes. For example, a

PROFESSOR is a STAFF member in a university. The class PROFESSOR is a subclass of the class STAFF. STAFF is a superclass of PROFESSOR. Inheritance of classes may simplify the system by reducing the number of independent system components that need to be identified. It also supports the interactive process of step-wise refinement with encapsulation and information hiding by allowing us to defer the definition of internal detailed activities of a class until necessary.

An object view of the world is a convenient one. Partitioning a problem domain into objects corresponding to a concept-oriented view of the real world counterparts is often more natural than a functional decomposition. Objects exhibit limited visibility, encapsulations of information, services and arguments. They additionally have a communication protocol to allow messaging between one another.

The models that are constructed with objects are actually mimicking the real world. Object orientation can model the real world of various systems. For example, Syntropy (Cook and Daniels, 1994), an object-oriented method mainly oriented towards software systems, suggests the construction of a series of three models to model the real world and a system to be implemented. The three models include a model of the real world (an Essential Model), a model of the system (Specification Model), and an Implementation Model that describes how the objects execute and send messages. Object orientation can contribute greatly to problem modelling, problem analysis and system design.

Object orientation technology mainly includes object-oriented analysis, object-oriented design, object-oriented modelling, and object-oriented programming. Object-oriented programming deals with the system implementation with programming languages, which are out of the scope of the study. Object-oriented analysis and object-oriented design are two basic techniques used in the object-oriented modelling of decision analysis and DSSs in the study.

1.7.2 Object-oriented Analysis

Object-oriented analysis (OOA) is a method of analysis that examines requirements from the perspective of the object model in the problem domain. It emphasises the building of real-world models using an object-oriented view of the world while traditional structured analysis techniques focus on the flow of data within a real world system. Actually, the model created by object-oriented analysis does not represent the real world, but models how it is seen given the semantics of the modelling paradigm. That is, the real world is modelled using objects, attributes, relationships, messages, and states. The model of a real world system, as an abstracted concept, is composed of classes, which are the abstracted representation of objects, together with their properties, relationships and messages.

1.7.3 Object-oriented Modelling

Object-oriented modelling offers a method that is capable of identifying the objects of a problem, understanding the structure and behaviour of each object, gathering all information and knowledge related to a particular object in one place, and showing how objects interact statically and dynamically. Therefore, in object-oriented modelling of decision analysis, a decision problem is analysed with object-oriented analysis. On the other hand, object-oriented modelling of DSSs comprises the analysis and design of the computer systems in object orientation.

1.7.4 Object-oriented Design

Object-oriented design (OOD) is a method of design "encompassing the process of object-oriented decomposition and a notation for depicting both logical and physical as well as static and dynamic models" (Booch, 1994) of the computer system under design. It focuses on the system design, which is about how to organise the construction of a computer system. Object-oriented decomposition divides a computer system as object clusters, from which logical and physical models of the system are constructed. Static and dynamic system models illustrate the system behaviour. They are represented by objects and their interactions. Object-oriented design uses classes and object abstractions to logically structure computer systems while the traditional structured methods use algorithmic abstractions.

In fact, there is no distinct demarcation between object-oriented analysis and design in the object-oriented modelling of a computer system. One can hardly tell when the analysis finishes and when the design begins. Some analysis results, such as the static and dynamic system models modelling the system behaviour, can be used as the output of the design phase. The deliveries from the object-oriented analysis are normally used as the input of object-oriented design, whose products can then be used as the blueprint for the computer system implementation by using an object-oriented programming language.

1.7.5 Advantages of Object Orientation

Object-oriented approaches have received wide attention in system development, especially in management information systems development, due to the advantages over the traditional structured approaches. Some examples show that object-oriented methods seem to be the only way to efficiently build systems such as distributed systems, user interfaces and workflow systems (Graham 1994). There is a widespread belief that object-oriented methods are better in many aspects than traditional approaches in the development of information systems. The future of object orientation looks very promising in that the object metaphor appears to be the most natural one to adopt in real practice. The principle benefits identified arising from the use of object-oriented methods are summarised as follows.

- Reusability and extensibility in software implementation are two practical benefits from object orientation. But they apply to analysis and design as well (Graham 1994). Reusability indicates that the system classes can be reused for the implementation of similar systems. Extensibility means that a system can be easily extended by adding classes as basic building components to the system.

- Well identified reusable classes that have been tested in the field on earlier projects in object-oriented systems are the basis for systems to be assembled, leading to high productivity and higher quality, thus better meeting business requirements.

- Object-oriented methods are a key tool for coping with system complexity, as is emphasised by Booch (1991). Partitioning systems on the basis of encapsulated objects help with scalability which can manage complex large system through scaling up from small to large. Class abstraction of the key elements of the problem domain, inheritance, aggregation based on real world objects and concepts also deal with management of complexity by modelling the world and its parts in a simple and transparent way.

- Object orientation offers a philosophy to naturally model the real world. Object-oriented analysis permits the system to be described in the concepts of the real world.

Because of these advantages of object orientation, it seems that there are many benefits obtained from the application of object orientation in modelling decision making and DSSs. As asserted by Graham (1994), the benefits at the analysis and modelling level of object orientation are potentially the greatest among analysis (modelling), design and programming. The system analysis specifications may be at least as reusable and extensible as object-oriented programs since the former is often less specific to a purpose. The next section introduces the modelling of decision making and DSSs.

1.8 Modelling of Decision making and DSSs for Natural Resource Management

Modelling can manage the complexity in decision problems and DSSs for natural resource management. It is futile to attempt to understand all the aspects of a decision problem and the decision making procedure and to implement an efficient and effective DSS without some form of framework or models. Furthermore, a decision problem, the decision making procedure, and its DSS are too complex to comprehend from a single perspective. This is because they can be comprised of several components, each defining an aspect of concern and representing a particular level of abstraction. In the past, these components were fragmented along functional lines, resulting in divergent purposes, discontinuous processes, and inefficient system implementation.

The models to be chosen can greatly affect the perspective of a decision problem and a DSS. A system of structured analysis may be viewed as a collection of algorithms and processes, with data flowing from process to process. In many decision modelling methods, a decision problem is rendered as a function to find a solution. However, in object orientation, a decision problem is modelled as a pool of individual objects interacting with each other, and a system modelled with object orientation is centred around a sea of classes and patterns of interaction that direct how those classes work together. Many other methods, as discussed in the following chapter, are also available to model decision problems, decision making procedures and DSSs.

Some characteristics of DSSs that differentiate these systems from traditional software systems make object orientation a promising method in modelling DSSs. Firstly, DSSs deal with semi-structured or ill-structured problems. This leads to the complexity for the development of the systems and heavy human-machine and human-human interaction in the systems. Due to its scalability, object orientation is capable of managing this complexity. As a simple yet natural modelling tool, object orientation will be able to model various interactions in the systems. Secondly, DSSs, especially those systems that support group decision making in natural resource management, have several system users with multiple levels. Besides, in DSSs, physical users such as decision makers and facilitators have to be modelled as internal classes as a part of a system instead of external entities in single-user software systems (A distinction should be made between the real world persons and their counterparts as modelled in the system). Object orientation can handle this multiplicity by producing different points of view from various users whose properties are encapsulated in classes. Thirdly, there are some classes which are intrinsic to DSSs, including decision makers, facilitators, criteria, objectives, decision attributes, alternatives, etc. That can be very useful for object-oriented modelling if due attention is paid on these intrinsic classes.

1.9 Objectives, Scope and Outline of the Study

This section describes the objectives of the study first. It also presents the reasons for using object orientation as the basis of the study. The limitations of the study are then listed. The organisation of the study is outlined at the end.

1.9.1 Objectives of the Study

Two major objectives are used as the motivation for carrying out the study. The first is to provide a philosophical methodology for decision analysis. According to Keeney (1992), a philosophical approach and methodological help is missing in most decision making methodologies to understand and articulate values and to use them to identify decision opportunities and to create alternatives. Keeney proposes a way to remedy that situation by focusing on values, which are the principles used for evaluation. These include ethics, desired traits, characteristics of consequences that matter, guidelines for action, priorities, value trade-offs, and attitudes toward risk. It is agreed that values are very important in understanding the problems and in identifying opportunities and alternatives. However, a broader point of view is needed to observe a problem and the problem context, to understand and articulate values about the elements of the problem, and to integrate all the related elements.

The second objective of the study is to find an effective and efficient way for decision analysis and DSS development in natural resource management. The decision analysis and DSS implementation in natural resource management is believed to be costly and time consuming. The study tries to create a conceptual framework for decision analysis and a DSS model, which can improve both effectiveness and efficiency of decision analysis and DSS development.

These two objectives are reached by the methodology proposed in the study, which provides a decision analysis framework and a DSS model based on object-oriented modelling and uses evaluation-principle guided analysis and design based on object orientation. The methodology can create models to represent decision problems and decision making procedures, and deal with the complexity and inefficiency in both decision analysis and DSS development while meeting the DSS principles.

Two observations have served as the impetus for the research on the object-oriented modelling of decision problems and DSSs. First, it is noted that the execution of decision analysis and DSS implementation tends to be an activity of low product output. The situation gets worse when it comes to complicated contexts such as natural resource management problems. As we know, even seasoned practitioners are repeatedly surprised by how much effort is needed to achieve useful results.

Increased use of object orientation may contribute to the productivity of decision analysis and DSS development in two ways. First, object orientation will be able to utilise the research outcome from the literature and the experiences from the previous case studies by the mechanism of reusing. It can be argued that a decision problem is usually unique and there will never be two identical cases, but no one can deny that human beings always make their decisions based on their knowledge and experiences. The point is how to use the knowledge and the experiences. Object orientation offers a mechanism, which is the inheritance of reusable classes, to reuse the existing proved knowledge and past experiences of a similar decision context. Secondly, object orientation provides a uniform tool to deal with almost all the aspects of decision making and DSS development. Most of the current methods typically cater to just one or two of the many phases of the total life cycle of decision making while the development of DSSs may be carried out with a totally different methodology. Users are forced to piece together a patchwork quilt of tools to deal with various phases and tasks as they arise over the life of a project. Object orientation will be able to allow these phases to be carried out in a uniform and coherent way.

The second observation as an impetus for the research on object orientation is that object orientation needs to be expanded to all main areas of DSS development in addition to some traditional aspects, to improve the efficiency of DSS development. Traditionally object orientation has been applied in several aspects of DSS development, including system programming with object-oriented languages, model management implementation with object-oriented methodology, object-oriented database module and human-machine interaction, which have been extensively used in other kinds of systems as well. In comparison with current methods, DSSs might be implemented more efficiently by creating a DSS model with object orientation. The reusable classes in the DSS model can be reused to efficiently construct a new DSS since the DSS model contains the DSS development knowledge and past experiences. Other knowledge can also be coherently integrated with the DSS model based on object orientation. For example, domain knowledge regarding the natural resource under consideration can be represented as classes of some decision elements. The procedure of implementing a DSS is that of instantiating (the action of identifying individual objects for a class) the existing super classes, and linking the objects obtained together in some way. A DSS, which is called WRC (Water Resource Commission) DSS, for water resource allocation in South Africa, has been developed in the study to demonstrate DSS development with the object-oriented approach.

It is therefore beneficial to apply object orientation in the modelling of decision problems, the whole procedure of decision making, and DSSs. Object orientation can facilitate decision making activities starting from problem analysis and ending with the implementation planning for the chosen alternative. It can also improve the productivity of decision analysis and DSS development. However, there are some limitations on the study.

1.9.2 Limitations of the Study

There are three main limitations on the study. First, the study falls in the problem domain of MCDM in natural resource management. This domain has been selected for study because of the complex features of natural resource management, which provide an excellent testing ground for development, and the experiences available in related projects at the University of Cape Town to carry out the study. It is possible, however, that not all of the conclusions may carry over to other domains, although it is believed that the results are widely applicable.

Secondly, only problem analysis and problem structuring are discussed in detail with regard to the application of the methodology proposed. This is because of the fact that these two aspects are the initial and most critical phases in decision analysis. It is also due to the fact that the later stages of decision analysis, i.e., evaluation and choice of decision alternatives, can be implemented as functions of the objects generated from the previous stages. Refer to Chapter 7 for more description.

Thirdly, this study does not include a comprehensive methodological verification for all the detailed ideas proposed. Nevertheless this study does explore the main theoretical aspects of the application of object orientation in decision analysis. The methodology proposed is primarily applied and tested in a hypothetical natural resource management case, which also tests the main functions of the DSS developed under the guidance of the DSS model proposed in the study.

1.9.3 Outline of the Study

The study is organised into nine chapters. This chapter introduces the basic concepts of related topics and includes the objectives and the scope of the study. Chapter 2 surveys the research in the literature on the application of MCDM in natural resource management, decision making processes and especially problem analysis and structuring, DSSs, the application of object orientation in decision analysis, especially in problem analysis and structuring, and in the development of DSSs. It also discusses the selection of an object-oriented method for the study and the evaluation principles of DSSs. Chapter 3 proposes the methodology and the philosophy based on object orientation for problem analysis, problem structuring, and DSS development. A methodological framework for decision analysis is proposed. The philosophy and methodology proposed in Chapter 3 are further described in Chapter 4. A conceptual framework for a decision problem macro system is presented in the chapter for the modelling of the problem solution procedure and the MCDM decision making procedure for natural resource management. Chapter 5 comprehensively captures the system requirements for DSSs for MCDM natural resource management decision problems, facilitated by the analysis of the DSS evaluation principles. Decision elements of decision problems and some other fundamental system components of DSSs are also extracted. In Chapter 6, the roles of the classes of decision elements and components of DSSs in the decision making of MCDM natural resource management decision problems are explored. A general system framework for DSSs for MCDM in natural resource management is generated in the chapter. Chapter 7 proposes an object-oriented MCDM method based on the ideas discussed in the previous chapters. Methodological guidelines for object-oriented MCDM in natural resource management are presented to provide basic approaches with which object orientation can be used in various decision analysis processes, including initial understanding, strategic analysis-brainstorming-decision element identification, structuring and exploring. Chapter 8 discusses the design of DSSs and the development of the DSS for water resources allocations in South Africa developed by using the existing DSS model. A hypothetical case study is carried out to evaluate the methodology proposed in the study. Questionnaires are used in the evaluation of the methodology and the system. Chapter 9 contains an overview of the study and some thoughts on the potential extensions and developments for future research.

Chapter 2

Literature Review

2.1 Introduction

This chapter contains a literature survey of the main relevant research topics of the study. These topics include the application of MCDM in natural resource management, decision making processes, problem analysis and structuring, DSSs, the application of object orientation in decision analysis, especially in problem analysis and structuring, and in the development of DSSs. The main focus of the present study is the development of DSSs to support multicriteria decision problems in a group setting.

There are three main emphases in the literature review. A first emphasis is placed on problem analysis and problem structuring since these are the initial activities in any decision analysis, whose success is largely dependent on the right identification and definition of the decision problem. They are usually regarded as the most critical phases in decision analysis. The second emphasis is the research on DSSs, which help decision analysts and stakeholders of a decision problem carry out various decision making activities including problem analysis and structuring. A third emphasis is on the application of object orientation in decision analysis and DSS development. This is because the methodology proposed in the study is based on the application of object orientation in the modelling of decision problems and DSSs.

It is generally found that a methodology is needed to provide techniques that can uniformly model decision problems and DSSs alike. Such a modelling methodology should be both philosophically and methodologically sound, being able to naturally model the real world in a simple and transparent way, to flexibly integrate other analysis techniques, and to make use of relevant knowledge and past experiences in decision analysis and DSS development. The modelling methodology should be able to bridge the gap between the DSS researchers (developers) and decision analysis practitioners to offer a common mechanism to understand decision making and DSS development. On the other hand, no object-oriented principles have ever comprehensively applied in the literature for decision analysis and DSS development even though object orientation has gained great success in information systems.

This chapter is organised as follows. Section 2.2 briefly introduces the application of MCDM in natural resource management. Decision making processes are described in Section 2.3. Section 2.4 reviews problem structuring and some characteristics of current methods. A suggestion for an ideal methodology for problem analysis and structuring is presented. Section 2.5 surveys the application of object orientation in decision analysis and especially in problem analysis and structuring. Section 2.6 examines the research on DSSs. The application of object orientation in the development and modelling of DSSs is studied in Section 2.7. The issue of object-oriented method selection is discussed in Section 2.8. Conclusions are contained in Section 2.9.

2.2 A Brief Review of MCDM Application in Natural Resource Management

For the last few years, MCDM has been applied vigorously in the support of decision making in natural resource management. Indeed, the well-developed field of MCDM offers a number of quite divergent methodologies to problems of natural resource planning (Stewart, Scott and Iloni, 1993). Talcott(1992) states that the various techniques of decision analysis, especially those dealing with multiple decision criteria, data envelopment analysis, and the analytic hierarchy process, provide reasonable ways to solve problems in environment management. Ellis (1992) adds that the environmental operational research projects have succeeded thanks to multi-objective methodology since environmental problems are fundamentally multi-objective in nature.

Decision making for natural resource management involves various stakeholders and communication mechanisms among them. Consensus is needed amongst the different groups or individuals who have conflicting social, economic and political interests. In such a complex problem as natural resources planning, there are issues that cannot be adequately modelled because they are qualitative in nature, unknown, or un-revealed by the decision makers (Brill, 1982). Hallefjord and Jornsten (1986) emphasise that the purpose of applying MCDM to long term planning problems is to explore solutions, to generate alternative strategies, and to gain insight into the problem, rather than to find optimum solutions.

Cohon and Marks (1975) discuss various MCDM techniques for water resource planning, including generating techniques which rely on prior articulation of preferences, and techniques which rely on progressive articulation of preference. Three criteria, i.e. computational efficiency, explicitness of trade-offs among decision alternatives and the amount of information generated for decision making, for evaluating MCDM techniques are suggested.

Romero and Rehman (1987) provide a comprehensive review of MCDM techniques, which include generating techniques, goal programming, ELECTRE, utility theory, mixed integer programming and compromise programming, in the application of planning and management of fisheries, agriculture, forestry and water resources.

Tecle (1992) claims that there are more than 70 MCDM techniques and at least 49 criteria to evaluate for choosing between them. In fact, a person's preference for a technique is probably based on individual experiences and the degree of familiarity with that particular technique rather than on its inherent superiority (Tecle, 1992; Hobbs, Chankong, Hamadeh and Stakhiv, 1992).

Schmoldt and Peterson (2000) suggest that group decision making with multiple values is becoming increasingly important in natural resource management and associated scientific applications. Multiple values are treated coincidentally in time and space, while multiple resource specialists and multiple stakeholders must be included in the decision processes. They further introduce a group decision making methodology in natural resource management and associated scientific applications. This methodology is claimed to be able effectively to deal with temporary, formal groups (e.g. workshops). It combines three components: brainstorming to generate ideas; the analytic hierarchy process (AHP) (Saaty, 1980) to produce judgements, manage conflicts, enable consensus, and to plan for implementation; and a discussion template (straw document). Resulting numerical assessments of alternative decision priorities can be analysed statistically to indicate where group member agreement occurs and where priority values are significantly different. An application of the methodology is made to fire research program development in a workshop setting.

According to Stewart, Scott and Iloni (1993), many previous applications of MCDM to natural resources planning were either on projects of a rather limited scale or not demonstrably implemented. Moreover, there is relatively little in the MCDM literature to deal with the process of generating alternative policies, although this is acknowledged to be an important and difficult task. Most MCDM approaches are based on the assumption that this task has already been completed, and that a well-defined set of alternative policies already exists. In addition, little reported literature has explored any general framework which addresses the particular problems in the application of resources planning. Examples of these particular problems are the model computational complexity for the consequences of specific policies and the need of the integration of subjective assessment of the less tangible problems with more objective data.

Decision analysis consists of several processes, some of which might play a more important role in the whole decision making procedure, especially in complicated MCDM such as natural resource management. The next section reviews some models of decision making processes in the literature.

2.3 Decision making Processes

Many suggestions about decision making processes have been reported in the literature. Perhaps the most well known one is by Simon (1965, 1977), who proposes the representation of the processes as: intelligence, design, choice, and review. The intelligence process involves searching the decision problem environment for various information and knowledge. The term "intelligence" is borrowed from military usage. It is the study and observation before action. The design process invents, develops and analyses possible decision alternatives and other decision explorations. In the choice process, a particular decision alternative is selected by using various skills for comparison and calculations. The review process is concerned with assessing the execution of a decision.

Mintzberg, Raisinghani and Theoret (1976) also present a famous empirically based model of decision making processes, in which the processes of decision making can be divided into identification, development, selection and choice. Identification, addresses the importance of identifying the correct problem. Development, which was viewed as the core of the decision making process, is to find "ready-made" solutions and intend to develop "tailor-made" solutions. Selection is concerned with eliminating alternatives which are not feasible. Choice involves bargaining, analysis, and judgement.

Adams, Courtney and Kasper (1990) identify several distinct and iterative stages of decision making, which typically include: decision problem identification, generation and analysis of potential solutions; decision alternative selection; decision implementation and subsequent (post-implementation) evaluation of the selected alternative.

Buede (1992) summarises decision making tasks as four basic iterative stages, i.e., problem definition, analysis, choice and implementation. Problem definition consists of the goal formation, information processing and problem structuring tasks. Goal formation enables the decision makers to recognise the problems with the status quo based on the objectives that have been either explicitly or implicitly defined. Information processing is to collect and analyse data and other information and to establish rules and relationships to describe reality. Problem structuring tasks include five main subtasks. The first subtask is the generation of alternatives in lieu of the status quo. The second subtask is the identification of evaluation criteria, which might be obtained by translation from the objectives defined, to enable decision makers to distinguish preferentially between the alternatives under consideration. The third subtask is the definition of uncertainties and dependencies that need better understanding before the decision is to be made. The fourth subtask is the specification of constraints that bind the alternatives. The fifth subtask is the definition of causal relationships between key variables that need modelling or understanding in light of the decision alternatives. The three tasks of problem definition may iterate to refine the definition of the problem.

The second phase of the decision process in Buede (1992), analysis, generates decision recommendations. Decision makers' subjective judgements are elicited; necessary statistical analyses are performed, logic-based reasoning is conducted, the expected utility of each alternative is calculated, and sensitivity analyses for critical variables are carried out. As part of the sensitivity analysis, the value of perfect information for uncertainties that have been explicitly modelled can be evaluated. Additional quantitative comparisons can be carried out for the direct comparisons of the expected weighted utilities for two alternatives on all of the objectives and comparisons of all of the alternatives on any two selected criteria. These two comparisons can enable substantial satisfaction on the preferred alternatives by the decision makers. New findings in the analysis phase may undoubtedly generate new insights for the problem definition phase, which might need to be revisited.

The third phase uses the same name – choice, and has a similar description as Simon's third element. The decision makers select an option out of the alternatives identified. Decision makers may need to convince at least themselves that the decision made is necessary and in their best interest (with non-conflicting objectives with other decision makers). This might lead to new questions that were either not fully examined or not even considered during the previous two phases of decision making. There might be a need for revisions to the problem definitions and decision recommendations.

The fourth phase, implementation, deals mainly with project control. The implementation agency establishes a schedule of tasks that must be completed to achieve the desired change from the status quo. Other optional elements of this phase may include identification of the critical activities and resources, definition of a task network, necessary modifications during the implementation procedure, drastic rethinking of the implementation plan, and possible re-initialisation of the whole decision procedure.

While Buede's (1992) decision making process model might be the most comprehensive one for decision making activities, Finlay (1994) proposes a decision making process model for the purpose of DSS study, which is claimed to be a "slightly modified and extended" version based on Simon's proposal. It consists of three phases, in which seven sub-stages are included. These three phases are structuring, understanding and action. The structuring phase contains two sub-stages of problem detection, which seeks to convince that the problem exists, and problem definition, which, as its name implies, defines the problem. The understanding phase includes three sub-stages of detailed DSS design, exploring courses of action, and decision taking. The implementation of the DSS helps exploration of action courses. Decision taking is the culmination of all the other stages of decision making since all the supportive work has already been done. The action phase is composed of two sub-stages of implementation of change and review. The change implementation and review are mainly concerned with the implementation and the assessment of the decision made. It seems that there is a limitation put by Finlay on the functional scope of DSSs. According to Finlay's model, a DSS is designed after the decision problem has been defined, which might be debatable in the decision analysis community.

Despite different viewpoints about the stages of decision making, there is "one constant" (Buede, 1992) in all decision process definitions. That is the belief that decision makers are constantly iterating amongst the stages of decision making, making revisions and bringing to their attention possible conflicts and inconsistencies in their preference as new insights are obtained and more knowledge about the problem is gathered in each interaction (French, 1984; Buede, 1992).

Among the stages of a decision making procedure, the most difficult and critical stage surveyed by Adams, Courtney and Kasper (1990) through questionnaires is the identification stage, i.e. how to clarify issues and relationship, and how to identify quantitative and qualitative variables. Problem identification is included in the definition of problem structuring in the next section, where problem structuring is a special term to indicate the initial decision making activities and covers most of the first stages of decision analysis in the reviewed decision making process models. Among the building tasks of DSSs, the most time-consuming stage is the model development stage since various decision models are needed to integrate into one system. Model development and management are out of the boundary of this study while problem structuring, including problem identification and problem modelling, is paid much attention to in the study.

2.4 Problem Structuring Methods

Problem structuring, generally also known as problem modelling, is an imaginative and creative decision making process of translating an initially ill-defined problem into a set of well-defined elements, relations, and operations (von Winterfeldt, 1980). Bana e Costa, Stewart and Vansnick (1997) suggest a detailed definition, which they believe might find wide consensus. According to them, the structuring and framing of a decision situation is a constructive and learning process. It seeks to build a more-or-less formal representation integrating the objective environmental components of the decision context, with the subjective and context-dependent points of view, concerns or objectives, in such a way that the value-systems of actors or stakeholders are made explicit. Problem structuring is considered an art since there is no attempt to uncover the scientific principles underlying the intuition and craftsmanship of the individual analyst (von Winterfeldt, 1980).

The basic structuring activities are identifying or generating problem elements (e.g., events, values, actors, decision alternatives) and relating these elements by influence relations, inclusion relations, hierarchical ordering relations, etc. The structuring process seeks to formally represent the environmental (objective) parts of the decision problem and the decision participants' (subjective) views, opinions, and values. Graphs, maps, functional equations, matrices, trees, physical analogues, flow charts, and venn diagrams are all possible problem representations. In order to be useful structures for decision analysis, such representations must facilitate the activities of modelling, elicitation, and numerical analysis.

Decision structuring, which occurs predominantly as analysis at a high strategic level, is essentially associated with complex situations, psychological constructions, uncertainty in input and output, and conflicts of individuals and strategies. To formulate a problem is primarily a conceptual and representational issue, while on the other hand, to solve a formulated problem is essentially a computational task. The most critical stage of problem solving, "a formal encoding of the problem under study" (Jaumard, Ow and Simeone, 1988) should be given more importance than the subsequent search stages. Problem formulation/ structuring, however, is still in its theoretic infancy (Corner J and Corner P, 1995) as shown by the general lack of their application discussion even though there are many methods available in the literature.

2.4.1 Characteristics of Existing Methods

There is a range of structuring methods or tools reported in the literature (Rosenhead 1989a; Corner J and Corner P, 1995). It is not the intention of this survey to introduce and examine each of these methods or tools. Attention is only paid to those of special interest even though all these approaches do make contributions in the area to various degrees. Some main methods reported in the literature include Objectives Hierarchies (Keeney and Raiffa, 1976), Strategy Generation Tables (Howard, 1988), Decision Trees (Holloway, 1979), Influence Diagrams (Howard and Matheson, 1984), Analytic Hierarchy Process (Saaty, 1980), Decision Analysis (Watson and Buede, 1988), Decision Conferencing (Phillips, 1988), Dialectical Inquiring Systems (Churchman, 1971), Idealised Planning (Ackoff, 1974, 1979), the LAMSADE School (Moscarola, 1984), Strategic Assumption Surfacing and Testing (Mason and Mitroff, 1981), Strategic Options Development and Analysis (SODA) (Eden, 1986, 1989), Soft Systems Methodology (SSM) (Checkland, 1989, 1990), Strategic Choice (Friend and Hickling, 1987; Friend, 1989), Robustness Analysis (Rosenhead, 1989c), Metagame Analysis (Howard, 1989), Hypergame Analysis (Bennett, Cropper and Huxham, 1989), AND/OR graphs (Amarel, 1967), and Structured modelling (Geoffrion, 1987, 1989).

Rosenhead (1981) suggests the six characteristics of the dominant paradigm of problem analysis and structuring in operational research. Among them are: "1) Problem formulation in terms of a single objective and optimisation. Multiple objectives, if recognised, are subjected to trade-off onto a common scale; 2) Overwhelming data demands, with consequent problems of distortion, data availability, and data credibility; 3) Scientisation and depoliticisation, assumed consensus; 4) People are treated as passive objects; 5) Assumption of a single decision maker with abstract objectives from which concrete actions can be deduced for implementation through a hierarchical chain of command; and 6) Attempts to abolish future uncertainty, and pre-take future decisions". The dialectical opposite of the orthodox approach is envisioned (Rosenhead, 1989a) to avoid those pitfalls and be able to illuminate decision making, design choice, and problem resolution in situations which cannot be fully structured in advance. Among the six characteristics for this alternative paradigm are: "1) Non-optimising, seeks alternative solutions which are acceptable on separate dimensions, without trade-off; 2) Reduced data demands, achieved by greater integration of hard and soft data with social judgements; 3) Simplicity and transparency, aimed at clarifying the terms of conflict; 4) Conceptualises people as active subjects; 5) Facilitates planning from the bottom-up; and 6) Accepts uncertainty, and aims to keep options open for later resolution". The paradigm alternative to the orthodox version is characterised by simplicity, transparency, flexibility, and openness in thinking, eliciting, and judging of various parties in problem analysis and structuring.

Some existing methods exhibit the characteristics of the alternative paradigm defined by Rosenhead (1989a) though not necessary all of the six aspects. For example, some use simple and transparent models, and others can explicitly clarify conflict. Among Strategic Options Development and Analysis, Soft Systems Methodology, Strategic Choice Approach, Robustness Analysis, Metagame Analysis, and Hypergame Analysis, Rosenhead (1989b) claims that most of the characteristics in the alternative paradigm of operational research are included to some extent in these methodologies. It is claimed by Rosenhead (1989b) that the ability to directly facilitate bottom-up planning remains out except in SODA where individual cognitive maps amalgamate into a single strategic map. Some methods, however, demonstrate other important advantages, which might not be covered in the six characteristics of Rosenhead's alternative paradigm. For example, AND/OR graphs (Amarel, 1967) are introduced to model strategy seeking problems involving "exogenous factors", such as uncertainty or conflict. It is embodied with a problem reduction or decomposition and heuristic features.

Influence diagrams (Miller, Merkhofer, Howard, Matheson and Rice, 1976; Howard and Matheson, 1979) provide a graphical structure for modelling variables and decisions and explicitly revealing probabilistic dependence and the flow of information. It is an intuitive notation for representing decision problems, and promises to have a significant impact on how decisions are modelled. They are well-defined mathematical structures and are also intuitively appealing. In addition, there are algorithms available to solve decision problems directly from the influence diagram representation. Influence diagrams can also naturally lend themselves to being generated automatically, making them the representation of choice for intelligent decision systems (Holtzman, 1989). More importantly, the concepts of influence diagrams can be used to capture the diverse knowledge and information possessed by an individual or a group (Howard, 1989).

Geoffrion's (1987) structured modelling provides formalism for model specification in which the structure and semantics of models are represented as hierarchically organised, acyclic, attributed graphs. Lenard (1993) notes that structured modelling has a lot in common with the object-orientation paradigm. Structured modelling formalises the notion of a definition system as a way of describing models. This is precisely what the object-oriented concept of a class and the class-composition graphs formalise. The model itself can then be regarded as a composite class having attributes each of whose domain is a composite class representing one of the modules.

Of late years, there has been a trend in the operations research and management science field to include a number of techniques termed "soft" operational research or "soft systems" approaches. The main purpose of these techniques is to facilitate individuals or groups to build up their understanding of a system in a structured and logical framework, but without forcing this structure into a rigid mathematical form. Particularly at the phases of problem structuring of decision analysis, soft methods are hailed as effective solutions (Rosenhead, 1989b). Three main soft methods include SODA (Strategic Options Development and Analysis) (Eden, 1986, 1989), SSM (Soft Systems Methodology) (Checkland, 1989, 1990) and the Strategic Choice Approach (Friend and Hickling, 1987; Friend, 1989). They can be used in a wide range of problem situations, without assumptions about the internal structure of the situations.

The SODA approach, based on the concept of "cognitive mapping", uses the traditional model building and analysis skills of operational research, but in an interactive and non-mathematical manner, to assist groups of decision makers involved with messy problems. Despite its OR pedigree, SODA employs cognitive mapping, which is not predominantly decision focused. Cognitive mapping is the basis of identifying organisational objectives and decision alternatives. Besides its emphasis on analysing the complex interactive process through which decision alternatives emerge, SODA focuses as much on organisational goals as it does on decision options (Rosenhead, 1989b).

The soft system methodology (SSM) aims at facilitating the obtaining of a general overview of the problem structure and interrelationships, as perceived by different actors. These perceptions can also be expressed with graphical devices (concepts and arrows) like SODA. With a background of the systems approach, SSM is concerned with how systems could work better.

While SODA and SSM are relevant particularly to the divergent idea of generating and learning phases of policy formulation, the Strategic Choice Approach is perhaps more oriented towards the analytical phase. In particular, it has emphasis on the identification of operational criteria for assessment and of needs for further research in the construction of decision alternatives. The Strategic Choice Approach focuses on the choice between candidate decisions.

Two special aspects of research in problem structuring are also reviewed, including the use of experiences in problem structuring and problem classifications, as they are primarily the basis for object-oriented modelling of decision problems.

2.4.2 Use of Literature and Experience in Problem Structuring

According to Simon (1973), when problems are ill-structured a great deal of effort and the ability to access a very large amount of potentially relevant information in the long-term memory is needed to structure the problem. To make use of literature and experience is very important in problem structuring.

MacCrimmon (1969) provides three approaches for problem structuring: (1) to examine the relevant literature, (2) to conduct an analytical study (for example, a model of inputs, process, and outputs), and (3) to perform causal empiricism (that is, to talk to the decision maker and the experts). Buede (1986) points out that the first approach is appropriate for some public decisions being decided by a governmental body for which a history exists. However, it is also noted that literature reviews are generally very inefficient and lack the necessary details compared to causal empiricism since the decision maker and the staff have so much on-the-job experience. For this reason, it is believed that causal empiricism is critical. It is the only one of these three approaches that can be used in isolation. The iterative use of causal empiricism and analytical studies can provide valuable insight into the value structure as well as other decision elements for complex problems. It seems that sufficient existing knowledge about the past is critical in using the first approach of problem structuring.

Decision models are actually representations of relevant knowledge in decision making including problem structuring. According to Muhanna (1993), there are different conceptions in the literature regarding the notion of a "model". Classical DSS literature treats models as computerised procedures to be managed. A second view of models, which is particularly reflected in proposals based on knowledge representation schemes, treats them as problem statements. Models have also been treated as data that are to be analysed or input to solvers. Yet in another interpretation of the model notion, models are viewed as systems which encapsulate their state and behaviour and present well-defined interfaces to the outside environment. In most circumstances, rather than regarding a model as a procedure or as data, a model is considered to be a problem statement (Chang, Holsapple and Whinston, 1993). For instance, the user's problem statement would include an explicit and complete specification of the constraints and objective function that are to govern a mathematical programming procedure's execution. However, decision models provide support for the alternative development and selection stages of decision making, but very little for the problem identification and diagnosis stages of the process (Adams, Courtney, Jr. and Kasper, 1990).

Keeney (1992) notices that there are numerous situations in which many decisions concerning the same general problem will be made over time. For an individual, examples include paying taxes and interpreting tax codes and regulations, purchasing automobiles or houses, and bargaining with the boss. For firms, examples include applying for a loan from a bank, introducing new products, and hiring key employees. For each case of a given general problem, the fundamental objectives are likely to be similar. However, important additional fundamental objectives in a specific case may concern what are new and will help you make better decisions in other problems of the same class.

Binbasioglu (1995) identifies two distinct modelling strategies: the employment of the basic decision making principles and the prototypical decision analytic structures. The latter is based on an intuitive belief that experienced problem solvers reach construction of a new system via modification of known systems. This suggests the possible employment of classification and constructive methods together at different stages of a problem solving process. Specifically, classification can be used prior to solution construction with the intention of discovering the problem components and identifying the pieces to be configured. The constructive method then adapts and integrates the identified model types.

Case based reasoning methods (Schank, 1982; Kolodner, Simpson and Sycara-Cyranski, 1985; Kolodner and Simpson, 1989; Liang, 1993; Dutta, 1996) have been suggested for modelling decision making. The idea is to formulate a new model by examining previous models of similar problems from a large library. A case is very similar to an object in object orientation, describing a significant chunk of knowledge with many attributes. Past models are stored in a structured manner as prior 'cases' and reasoning methods are used to identity those that bear similarity to the current problem situation. This approach views model formulation as one of making incremental changes to an existing but similar model. The case based approach assumes that human experts also use the analogical approach in their modelling activities.

2.4.3 Classification in Problem Analysis and Structuring

"Taxonomies" of problems are found useful for problem identification (von Winterfeldt, 1980). The taxonomies should lead an analyst to a class of problems which have characteristics similar to the decision problem under investigation. Unfortunately, according to von Winterfeldt (1980), the existing problem taxonomies are ill-suited for this purpose because they use mainly analytic categories to distinguish problems. Such categories are derivatives of the decision analytic models and concepts, rather than characteristics of real world problems. von Winterfeldt (1980) suggests that substantive rather than analytic characteristics identify real problems. Substantive characteristics are generalised content features of the problems belonging to the respective class. To become useful for problem identification, taxonomies need to include such substantive problem characteristics.

Von Winterfeldt (1980) further contends that it would be helpful for the analysts of decision problems to make an effort in addressing the question of generalisability when modelling a specific problem, and in extracting those features of the problem and the model that are transferable. Not every decision analysis can afford to be as broad and time consuming as the previous study, although decision analysis usually has a much more specific orientation towards producing a decision rather than developing a generic structure. An inductive approach of generalisation could be coupled with more research-oriented efforts and with examination of similarities among past applications. This middleground (between too specific and too general models and structures) would be filled with prototypical structures and models rather than filling it with analytically specific but substantively empty structures and models. Requirements for prototypical structures for four typical classes of decision problems (siting, contingency planning, budget allocation, and regulation) are discussed in von Winterfeldt's paper.

Decision "templates" of generic problem structures are suggested by Keller and Ho (1988). Use of these templates is regarded as one promising general approach to problem structuring. Decision templates of generic problem structures are then augmented with information from a specific problem, as suggested by Weiss and Kelly (1980) and Weiss (1985).

Decision Class Analysis (DCA) proposed by Holtzman (1989) is about analysing and designing a class of similar decisions as a single unit at a sufficiently high level of abstraction. It is claimed to be able to greatly reduce the overall expense and time typically associated with professional decision analysis. To do this, relatively sophisticated techniques such as knowledge engineering must be used. (Holtzman, 1989) contends that the concept of analysing a class of decisions can be naturally described in terms of a rule based system. The goal of the rule-based system embodying a decision class analysis is to generate a fully assessed Well-Formed Decision Influence Diagram (WFDID) representing the decision problem. The rules in a decision class analysis can propose decision options that have not been previously considered. Kim and Park (1997) apply neural networks in building influence diagrams with decision class analysis. They suggest a neural network to implement decision class analysis. Influence diagrams are again employed to represent the decision problem.

Decision domain analysis (Arango and Prieto-Diaz, 1991) has been defined as the process of identifying and organising knowledge about some classes of problems in a problem domain, to support the description and solution to those problems. Decision domain analysis technique is used in (Wang, 1995) to determine the requirements for resources in supporting certain types of decisions as well as the possible paths in such decisions.

According to Pomerol (1997), the first phase of decision making is to find one or several recorded states or situations of a problem similar to the perceived current state or situation. Depending on their context and complexity, this operation can be denoted "pattern matching" or "diagnosis". Classification is diagnosis when the current states can be enumerated. Classification must be made according to many attributes or parameters. The rough set theory (Pawlak, 1982, 1991; Pawlak and Slowinski, 1994a,b) is suggested as one of the most promising new methods. The rough set theory was originally proposed by (Pawlak, 1982) to deal with representation and processing of vagueness, imprecision and uncertainty. In a certain sense, it refers to machine learning, knowledge discovery, statistics and inductive inference (Pawlak, 1997).

The rough set philosophy is based on the assumption that every object of the universe is associated with a certain amount of information, e.g., data and knowledge. Objects characterised by the same information are indiscernible or similar in view of the available information about them. The indiscernibility relation generated in this way is the mathematical basis of rough set theory (Pawlak, 1997). It induces a partition of the universe into blocks of indiscernible objects. Any set of all indiscernible or similar objects is called elementary set, which forms a basic atom of knowledge about the universe. Any subset of the universe may be expressed in terms of elementary sets either precisely as a union of some elementary sets, or approximately only. In the former case, the subset is referred to as a crisp set, while in the latter case, it is referred to as a rough set. Consequently each rough set has boundary-line objects which cannot with certainty be classified as members of the set or its complement with the available knowledge. This leads to the view that knowledge has a granular structure and some of the concepts of objects of interest cannot be discerned and appear as the same or similar. Therefore, a vague concept is replaced by a pair of precise concepts – the lower and the upper approximation, which are two basic operations in the rough set theory, of the vague concept. The lower approximation contains all objects which definitely belong to the concept and the upper approximation consists of all objects which possibly belong to the concept. The difference in between constitutes the boundary region of the vague concept. In this way, the two basic operations classify data and knowledge into fundamental patterns.

Use of knowledge and experiences is very important in problem structuring. Classification offers a promising means to represent and retrieve various information for decision problems. However, there are still some issues which need further exploration even though much effort has been put into the research of problem structuring in various aspects.

2.4.4 Issues and Perspective

There are some restrictions on many current methods for problem structuring. One of the evident restrictions on many formal methods is that the analytical mathematical forms (both overtly and covertly) make the analysis inaccessible to most of its potential users (the so-called "soft" method can cope with this issue in some sense). Graphical methods such as diagrams can display quite intricate networks of influence, complexity and understanding of a situation. Another restriction is that some methods force the problem into a particular quantitative mould. By contrast, it must be possible for decision makers to identify events or outcomes without being obliged to use numbers to express significance. Thirdly, in the context of a single organisational structure, some structuring methods such as Strategic Options Development and Analysis

(SODA) may be quite developed. However, in contexts such as national or regional resources planning, it is not clear how they can be carried out as different parties may be dispersed geographically and culturally in these situations (Stewart, Joubert, Scott and Low 1996). These issues do not fully correspond to the group work situations which may be dealt with to some extent by Soft Systems Methodology (SSM).

The second issue in problem structuring is that few methods reported in the literature encompass the four streams of thought on problem structuring. Woolley and Pidd (1981) in their literature review of problem structuring propose these four main streams of problem structuring thoughts. The first is the checklist stream, which contains a step-by-step procedure for problem structuring. The second is the definition stream, which involves the definitions of the problem in terms of client, objectives, alternatives and measures. Thirdly, there is the science research stream, which focuses on gaining an understanding of the problem situation. Finally, there is the people stream, which regards the definition of the problem as a function of people's perceptions. Although these are only one possible grouping of clusters of ways of thinking about problem structuring in the literature, each group represents a coherent, if by itself inadequate, understanding of the process of problem structuring. It is believed therefore that it is methodologically advantageous for a method of problem structuring to comprise all these four streams.

The problem of low productivity in decision analysis is rarely addressed in the literature. Geoffrion (1987), however, lists the low productivity in MS/OR (management science/operations research) as a lamented problem confronting the MS/OR community. As a division of operations research, decision analysis, especially decision problem modelling, has the same problem. Von Winterfeldt (1980) hints that modelling a specific problem and extracting those features of the problem and the model could be so inefficient that in some circumstances it will not be affordable as it is broad and time consuming. Geoffrion (1987) identifies the four factors which may contribute to low productivity. The first factor is the interface problem which presents a model in a format acceptable to the chosen solver. The second factor is the model integration problem arising from the fact that most software addresses just one among many kinds of models that can arise. The third factor is the completeness problem of software that typically caters to just one or two of the many phases of the total life-cycle of analysis. The fourth factor is the problem of problem representations. The first three might be easily dealt with respectively by creating interface standards between models and solvers, integrating various models into a system, and containing more phases of analysis within a system. The last contributing factor still remains to be solved in terms of broad MS/OR and decision analysis specifically as well. Geoffrion (1987) contends that at least three distinct representations are typically used for modelling: a user-friendly representation for communication among different people, a mathematical representation for analysis, and a computer-executable representation for computation. These inconsistent and redundant representations are inefficient since they demand many skills from users.

Most important is the issue of philosophy and methodology. Keeney (1992) points out that a philosophical approach and methodological help is missing in most decision-making methodologies to understand and articulate values and to use them to identify decision opportunities and to create alternatives. Value-focused thinking is suggested as the way to remedy that situation as values, which are principles used for evaluation, are what matters in decision making and should be focused on. Values include ethics, desired traits, characteristics of consequences that matter, guidelines for action, priorities, value trade-offs, and attitudes toward risk, all indicate values.

It seems that any problem structuring method should be sound in both methodological and philosophical terms. It should be able to naturally model the real world in a simple and transparent way and allow the problem to be observed from different viewpoints. Problems should be able to be analysed in flexible ways. The method should be able to integrate "soft" and hard problem analysis approaches and comprise the four main streams of problem structuring thoughts proposed by (Woolley and Pidd, 1981). It should be able to be applied to a wide range of problems, especially those involved with multiple parties dispersed geographically and culturally. The problem analysis and structuring should be able to be guided by "taxonomies" (von Winterfeldt, 1980) and with the help of experience and relevant knowledge.

Many methodologies such as Soft Systems Methodology (SSM) and Fuzzy Sets (Zadeh, 1965; Greco, Matarazzo and Slowinski, 1999) have been applied in problem structuring and decision analysis so as to effectively and efficiently support decision making. Object orientation is a promising philosophy and methodology to be applied in this field. The next section reviews the application of object orientation in problem structuring and decision analysis.

2.5 Object Orientation in Problem Modelling and Decision Analysis

There are many general applications of object orientation, mostly included in two divisions. The first division includes those where much has been achieved, such as user interfaces, hypermedia, multimedia, distributed and client/server systems, geographical information systems, simulation, process control, and artificial intelligence. The second division includes those where a potential of success exists, such as data processing, database, executive information systems, and business modelling (Graham, 1994).

While object orientation has gained great success in many areas as listed above, no object-oriented methodological principles have been seriously proposed or discussed in the literature for problem structuring and decision analysis. Little research has been conducted on the application of object orientation in decision analysis and problem structuring.

Objects and relationships between objects are used by Belardo and Harrald (1992) to represent knowledge of problem structuring for decision problems of planning for catastrophic events. When a knowledge base is first created, experts are asked to describe the objects they discussed during the hindcasting session and the decision structuring exercise. These objects can be identified as stakeholders, events, activities, resources, or other categories considered appropriate by the experts. Each object has associated with it characteristics or instances called attributes. These attributes can be initialised with particular value. Each object is also connected to other objects through relationships. A diagram is used to illustrate the use of the object-oriented representation tool for representing the knowledge base required for response to an oil spill problem. In the crisis domain, man-made crises are a subclass of the general category crises, and the oil spill is a specific type of man-made crisis. The federal and state response organisations, natural resource trustees, and environmental and economic interests represent stakeholders who are affected by such a crisis. But, no object behaviour of object orientation is discussed in the method, which may be termed intuitive adoption of object orientation.

Graham (1994) gives a case study to show how an object-oriented method is applied to modelling the structure of an organisation aiming to provide a business model that could be used to simulate different organisational strategies. It is contended that object-oriented analysis can be applied to organisational strategy as well as to conventional systems development.

It is noticed that there are very few publications found in the literature about object orientation in decision analysis and problem structuring. However, there is a need for research in object-oriented decision analysis and problem structuring since object-orientation may be beneficial to the field of decision analysis. For example, object orientation may be able to support efficient and effective representation and elicitation of decision elements, such as alternatives, preferences, and objectives. It may also be able to build a shared understanding amongst members of the decision making concerned, as to the dynamics of the circumstance, the goals to be attained, and the options available.

Object orientation can not only be applied in decision analysis and problem structuring but also in the modelling and development of DSSs, which support various decision making activities. In the next section, the development of DSSs is reviewed.

2.6 DSSs

This section mainly surveys the structures and the development of DSSs. Special attention is paid to on GDSS. Some issues are found in the current research.

2.6.1 DSS Components

Sprague and Carlson (1982) suggest three constituents: database management (DBMS), model base management (MBMS), and dialogue generation/management (DGMS) for a DSS' software system. Database management deals with the relevant data and other information for the decision problem and decision making activities. Model base management handles the representation and usage of decision models, which are abstracted representations of decision problems, usually expressed in mathematical terms. Dialogue generation/management is the interface for decision makers to interact with the system.

Bonczek, Holsapple and Whinston (1981) allow all aspects of the SC (Sprague and Carlson, 1982) framework in their own framework, BHW (Bonczek, Holsapple and Whinston, 1981). The BHW framework has three major components: a language system (LS), a knowledge system (KS), and a problem processing system (PPS). The language system interacts with users for information input and output. The knowledge system keeps relevant knowledge that is useful for resolving the decision problem, while the problem processing system uses various data and knowledge to solve the decision problem.

Bidgoli (1996) suggests the general components for a GDSS. A GDSS comprises software, hardware and people (facilitators and other decision making participants). The software components may include database and database management capabilities including internal and external data; modelling capabilities including mathematical and statistical models; dialogue management with multiple user access including menus, GUI (Graphical User Interface) and command driven interfaces; and specialised application programs to facilitate group access. The hardware components of a GDSS may include general purpose I/O (Input/Output) devices (dumb terminals, personal computers, workstations and voice I/O); central processing unit; shared viewing screen (for the group) or individual monitor for each participant; network systems which connect different sites and participants to each other. Decision making participants can use their terminals dispersed in physically different places to communicate with each other and see each other's response through the shared or individual viewing screen. The facilitator uses a dialogue management system to assist them to utilise the GDSS.

Multiple terminals should be integrated into the structure of a GDSS so that individuals in a single workshop (group), or in different interest groups, can develop their own preference structures in parallel, before coming together to assess potential compromises (Stewart, Scott and Iloni, 1993). The technological feasibility of this approach is well illustrated by the Co-oP system (Bui, 1987), which is implemented with integration with particular MCDM methods as a general framework for GDSS.

Model base and its management, which is also known as model management together, are thought as a means of knowledge and experience reuse in problem solving by Blanning, Holsapple and Whinston (1993). According to them, a salient component of model management will be experience-based problem solving, in which a decision maker "solves new problems by adapting solutions that were used to solve old problems" (Riesback and Schank, 1989). The main task of decision model management, and information management more generally, is the identification and integration of decision models and other information types to solve non-routine problems. People usually solve these problems in part by deduction from first principles, which include knowledge of the problem domain coupled with simple logical rules and heuristics, but largely by recalling and making productive use of previous problem-solving experiences. This would require that model management maintain libraries of successful and unsuccessful problem-solving efforts and present interface procedures for identifying appropriate analogies in the form of models and other sources of information relevant to a particular problem.

Antunes, Alves, Silva and Climaco (1992) gives an example of a model base in their decision support framework, which includes an integrated MOLP (Multiple Objective Linear Programming) interactive method base with a dialogue base and a data management module to support the decision maker (DM) to explore the problem. The interactive MOLP base offers the possibility of method switching at any interaction with the decision maker to make the most of the potentialities of each method and to support a learning process.

Besides a user interface, a problem processing unit, a model base, a data base and a knowledge base, DSS components may include other elements. Among these are a case base for keeping old problem cases, a graph base for displaying various graphs to users, and even an audio-visual resource base brought by the multiple media technology to present sound and image to users. These components are listed for the purpose of completeness.

2.6.2 GDSS

Desanctis and Gallupe (1987) propose three distinct levels of decision making support for group support systems. The purpose of a group support system at the first level is to improve the decision processes by removing common communication barriers. Most GDSS and EMS (Electronic Meeting Systems) fall into this category. Group support systems at the second level are additionally to permit groups to work simultaneously on problem identification and solutions while viewing their analysis. Examples are GroupSystems (the University of Arizona, Ventana) and SAMM (the University of Minnesota). Group support systems at the third level use expert system (ES) and artificial intelligence (AI) technologies to provide advice in selecting the most suitable rules for enhancing group discussions. Argnoter is an experimental example of such a system.

Bui (1987) clearly identifies three important phases in a group decision making procedure as follows: 1) Generalised and unified decision support for individual decision making, 2) Communication support, and 3) negotiation support to assist the individual in negotiating with other decision makers of the group. Each individual interfaces with a personal DSS which facilitates the processes of formulating the individual's own preference rankings on a set of decision alternatives. Thereafter, the value judgements can be communicated between individuals and/or to a facilitator. Consensus is reached through communications and negotiation based on individual perceptions on the problem and individual judgements.

Bui (1987) also gains some experiences in developing a GDSS, Co-oP. It is suggested that GDSS should be distributed, loosely coupled and process driven. A distributed and loosely coupled architecture of DSSs is contended to be able to provide autonomy and flexibility for individual decision making, and homogeneity and simplicity for group problem solving. Also, process driven DSSs are claimed to be able to deal with the unpredictable nature of group problems. This is because collective decision processes have been shown to be the only elements in a DSS that are stable enough to fit into most collective problems, reasonably structured to be implemented, and sufficiently controllable to guarantee appropriate use.

Phillips (1988) discusses features that a GDSS should possess. Among them are the ability to help structure the thinking of the group, the capacity to deal with group dynamics, understandable methods of operation to participants, and flexibility. Such a system will be able to support communications and interactions among decision participants in flexible ways, and also be able to do problem structuring for the group.

Vetschera (1990) surveys recent developments in GDSS for individual and group support. Typically, group decision making involves both individual and group stages. The obtaining of a group opinion is classified in two ways, either the whole group participates in forming a new opinion, or individual opinions, which were previously obtained, are aggregated. Aggregation techniques are often based on MCDM methodology, which can also be used to determine compromises between group members. Vetschera concluded that there are no systems which provide interactive guidance, i.e. providing a structure for the decision processes without determining the outcome, at the individual stage, and for facilitation at the group stage.

Ackermann (1994) compares two major schools in the research of GDSS. Considerable attention in recent years, especially in the United States, has been paid to computer-network based GDSS, while in the United Kingdom manual methods are still popular. The author discusses some issues of concern around aspects of GDSS design, including the constraints of fixed location for hi-tech rooms, the lack of flexibility of specially designed physical environments, levels of participation in data capture, presentational difficulties, managing the inherent complexity of large volumes of data, the degree of control by the team leader/client manager in relation to the control by the chauffeur/facilitator, and behavioural impacts (the management of group dynamics, the nature of conflict, and the inability to cater for informal as well as formal languages). The conclusion is that some combination of the two schools together with the partially computerised systems could provide groups with the maximum support.

Venkatraman (1996) extends previous GDSS research to develop a framework for future GDSS research and design. Six situational factors are identified, namely group size, member proximity, task type, group environment and group development stage, to describe a given group decision making situation. The features across different GDSS are standardised into different levels of GDSS features. A multidimensional framework is then presented for conducting future GDSS research and it can help GDSS development by identifying the appropriate GDSS features for the appropriate group decision situation.

2.6.3 Development of DSSs

Radcliff (1986), Golden, Hevner and Power (1986), and Taylors (1987) surveyed some of the decision software packages available at that time such as Decision Modelling, RiskCalc, Consultant, Decision Master, The Confidence Factor, Rank Master, MacChoice, Weighted Point Rating, and Priority Decision System. Other examples, which fall within the four categories defined by Buede(1992), include structuring packages (Cope, Stella for Business),Value Matrix packages (P/G%, Best Choice, Lightyear), decision tree packages (DATA, Supertree), multi-attribute packages (Arizona State University, EXPERT 87), and inference packages (Decision Factor, Decision Power). But these packages often provided little functionality more advanced than an evaluation matrix (Buede, 1992).

Keen (1987) suggested a shift in the definition of DSSs from improving the effectiveness of managerial decision making in semi-structured tasks to applying artificial intelligence and other computer techniques to improve creativity and learning during decision making. Creativity means new findings in many aspects including new perceptions about the decision problem, introduction of new mechanism of action to achieve the objectives, and more decision alternatives.

Since 1990, great advances have been seen in the decision support software market (Buede, 1996), especially for model structuring, but support is still limited. There are several new packages coming into the market every year, although the majority are upgraded and improved ones with some novel features. For instance, Criterium and HIVIEW have a brainstorming technique for the user to input objectives and graphically build them into a hierarchy. Other examples include Expert Choice, Decision Explorer, and VISA. However, present software developers continue to focus on analysis features rather than support for non-analysts to structure the problem and a more complete set of elicitation support. Common users are hardly qualified and comfortable to use these packages (Buede, 1996), but specific examples are not given.

Reusable DSS modules are proposed by Nunamaker, Applegate and Konsynski (1988) Generalised DSS architectures integrate the data, dialog and modelling components of a DSS and provide a framework for DSS design. Static system components can be designed and do not change from one application to another. This forms the basis of a library of reusable DSS modules (Nunamaker, Applegate and Konsynski, 1988).

Berztiss (1998) proposes a two-level domain model for flexible DSSs influenced by the very rapid change of the environment in which they are embedded, and also by the increased dependence of organisations on externally developed systems. A generic base model remains for the most part unchanged over the time, and an upper level model consists of specialisation for specific contexts. The two-level modelling can make a DSS easily adaptable to environmental changes (context evolution) and the changes of external systems (context switching).

Aiken and Liu Sheng (1991) suggest representation uniformity and inheritance properties for analysing DSSs. A method called SES (System Entity Structure) is used for the specification of GDSS boundaries in a way much like object orientation. Specification entities inherit the aspects and attributes of parent entities. The authors contend that uniformity of representation and inheritance properties of the SES ensure a consistent and concise representation of the problem domain.

Dyer, Fishburn, Steuer and Wallenius (1992) view recent research as a "healthy trend" to focus on finding "simple, understandable, usable" approaches which will undoubtedly be practical in building DSSs, and on supporting all the decision stages from problem structuring to solution implementation.

The first important task for the implementation of a specific DSS is to outline the deficiencies in the present decision making (Evans and Riha, 1989). Prior to the system analysis and design, software evaluation and selection for DSS generators should take place (Le Blanc and Jelassi, 1989). Moreover, it is preferable to develop a demonstration prototype at first. Using a DSS generator for the development of specific DSSs will be advantageous (Le Blanc and Jelassi, 1989). According to them, it will reduce personnel requirements and development costs, such as the "number of necessary maintenance changes, activation, testing effort, and the number of bugs". In fact, most specific DSSs are being developed with general-purpose DSS generators such as FOCUS, IFPS, and Lotus 1-2-3.

Eom and Lee (1990) conduct a survey of DSSs and their applications over the period of 1971 and 1988. A total of 203 articles are compiled and classified according to several different application areas including agriculture, education, government, hospital and health care, military, natural resources, urban and community planning, corporate functional management, and others. The survey strongly indicates that computer-based DSSs are increasingly applied in profit and non-profit organisations such as governments, the military, health care, and education. In corporate functional management fields, marketing, transportation and logistics contain the largest number of application articles, followed by production, operations management, finance, strategic management, and human resources management.

Eom, Lee, Kim and Somarajan (1998) review DSSs and DSS applications over the period of 1988 and 1994. Two hundred and seventy-one publications are included. This survey clearly indicates that DSSs are increasingly being implemented in many organisations and that there have been significant financial and non-financial benefits of DSS applications. The majority of DSSs are being applied to support operational and tactical decisions. The systems surveyed are equipped with a variety of tools such as graphics, visual interactive modelling, artificial intelligence techniques, fuzzy sets, and genetic algorithms.

Development of specific DSSs in various fields is facilitated by a host of new tools including database management, statistical analysis, and commercially available DSS generators.

2.6.4 Some Examples of DSSs

There are many DSSs currently available on the market. The intention to list some examples here is to demonstrate the development in the field. It is not intended to give a comprehensive survey of all the systems. Most proprietary systems (Eom and Lee, 1990; Eom, Lee, Kim and Somarajan, 1998) are not included here.

(1) VISA(Visual Interactive Sensitivity Analysis)

VISA (Visual Interactive Sensitivity Analysis) (Belton, and Vickers, 1989; Visual Thinking International, 1994) uses multiple criteria analysis to provide extensive support to evaluation and sensitivity analysis. Decisions are modelled using hierarchically weighted value functions. An important and distinctive feature of VISA is its extensive facility for visual interactive sensitivity analysis, which enables decision makers to explore the implications of changing or differing priorities and values. It is not, however, designed to facilitate the process of problem structuring. VISA can define and evaluate multiple criteria not necessarily directly linked to quantitative attributes, and allow direct scoring of the alternatives in terms of these criteria on a 0-100 scale. VISA offers some graphical representation such as bar graphs, thermometer, and profile diagrams for flexible analysis. VISA is a valuable aid to obtain consensus between conflicting parties by using the interactive modification of weight bars and the side-by-side display of multiple "thermometer" scales.

(2) Expert Choice

Expert Choice (Export Choice Inc, 1995; Fernandez, 1996) takes the AHP (Analytic Hierarchy Process) methodology (Saaty, 1980) to structure a multi-attribute decision problem and to evaluate the relative desirability of alternatives. Expert Choice is flexible for assessment in several ways such as graphs for rating, data input, what-if analysis, and pairwise comparison of alternatives. A group version, Team Expert Choice, is available to help multiple users to make collaborative decisions with the use of hand-held radio keypads for voting.

(3) HIPRE 3+

HIPRE 3+ (Hämäläinen and Lauri, 1993) is a decision support software product integrating AHP (Analytic Hierarchy Process) and SMART (Simple Multiattribute Rating Technique) (von Winterfeldt and Edwards, 1986) for decision analysis and problem solving: The two methods can be run independently or be combined in one model. Its group version, HIPRE 3+ Group Link, is a group decision support software which combines individual prioritisations given by the Analytic Hierarchy Process (AHP) into an interval AHP model called preference programming model. The full GDSS consists of two softwares, HIPRE 3+ and HIPRE 3+ Group Link. Group members make their AHP prioritisations with HIPRE 3+, after which the models are combined with HIPRE 3+ Group Link.

Web-HIPRE (Helsinki University of Technology, 1998) is a web-version of the HIPRE 3+ software for decision analytic problem structuring, multi-criteria evaluation and prioritisation by using Java-Applets.

(4) Co-oP

Co-oP (Bui, 1987) comes as an excellent GDSS. Co-oP is implemented with a high level of technology, which should be of interest for the structural design of GDSS. It differentiates the within-interest and between-interest consensus-forming phases. Co-oP does not, however, identify the concepts of attributes as distinct from criteria of evaluation, nor does it identify policy elements, for the individual and group support. Co-oP seems to be primarily dedicated to the cases in which group members have a substantial degree of conformity of purpose (e.g., the management team of a company in the process of making a senior appointment) other than those cases with high levels of conflict and antagonism, such as resources management.

Co-oP runs on a network of individual workstations. It contains a set of MCDM methods, techniques of aggregation of preferences, and a consensus seeking algorithm to support negotiation. Electronic communication among group members is monitored by a group norm filter which is adaptable to a large number of collective decision situations.

(5) HIVIEW

HIVIEW (Barclay, 1987; Enterprise LSE Ltd, 1998) is a decision support tool using multi-criteria analysis as the method of choice for government and business. It enables users to structure their options and the criteria upon which options are differentiated, thus providing an audit trail of the decision process. HIVIEW allows definition of different criteria of evaluation not necessarily directly linked to quantitative attributes, and also allows direct scoring of the alternatives in terms of these criteria on a 0-100 scale. In fact it literally displays a "thermometer" scale on which the scores can be indicated and changed on-line by using a mouse.

(6) EQUITY

EQUITY (Barclay, 1988; Enterprise LSE Ltd, 1998) also uses multicriteria analysis to help organisations determine optimum use of resources among a variety of expenditure items, such a competing projects, purchases, or system components, mapping many different scenarios and can help bid analysis and project prioritisation. It can also be applied in other cases where the user has to choose one from many options but has few criteria on which to differentiate those options. EQUITY might be applied in a simplistic context of dividing a single resource between competing users. It seems to be relevant to special resources allocation problems, for example, allocation of fixed resource of water in times of drought, rather than to the general resource planning problems.

A unique feature of EQUITY is that it requires the specification of general societal criteria of evaluation. The evaluation of each scenario by the user group can be done on a "thermometer" scale in terms of their contributions to each of these criteria. The weight assessment of different contributions can be calibrated between the societal criteria and between user groups for each societal criterion.

(7) GroupSystems

GroupSystems Software (Ventana Corporation, 1994) is an electronic meeting support software that supports group processes such as brainstorming, list building, information gathering, voting, organising, prioritising, and consensus building. It contains a collection of tools to allow a group of people to work interactively at separate workstations. Individual decision makers enter their ideas at their own terminals simultaneously and anonymously into a list of ideas contributed by other participants, which appears on all the participants' screens. Participants can further comment on any of them

(8) Decision Explorer

Decision Explorer (Banxia Software Ltd, 1996), formerly called Graphics Cope, is based on a "causal mapping" technique, which is also known as "cognitive mapping", a method to create a model of the world graphically using concepts and links. Decision Explorer facilitates the modelling and management of thoughts and ideas around problems to provide prompts and analysis. Decision Explorer allows support of many mapping techniques for use both by a single user and in group work. Map analysis is provided to explore the problem, bringing about thoughts, discussions and solutions.

(9) ELECTRE III/IV

ELECTRE III/IV (Lamsade, 1994) uses the outranking methodology (Roy 1973, 1990) for the construction of binary imprecise relations to model and explore the decision makers' preferences in order to construct a partial pre-order defined on a set of alternatives.

(10) Logical Decision

LDW (Logical Decisions for Windows) (Logical Decisions, 1989; Smith, 1996) implements MAUT (Multi-Attribute Utility Theory) analysis, and integrates the tradeoff, SMART, SMARTER and AHP methods for weight assessment. Logical Decision stands in those packages which are technically sound but have to take the assumption that the criteria of evaluation are exhaustively identified and linked to quantitative system attributes. This assumption, however, does not seem to be applicable in complex decision problems such as natural resource management.

(11) Sensitivity/Supertree

Sensitivity/Supertree (Mcnamee and Celona, 1992; Strategic Decision Group, 1996) use decision trees and influence diagrams as the basic tools for decision analysis. Sensitivity identifies critical uncertain inputs to spreadsheet decision models, and

Supertree takes into account the critical uncertainties. In combination with spreadsheet models, they automate the decision analysis processes, helping the users to generate insights that lead to better decisions. Sensitivity/Supertree defines different shapes of figures for uncertainty, decision, value and information flow in building influence diagrams.

(12) CDP (Criterium Decision Plus)

CDP (Criterium Decision Plus) (InfoHarvest Inc, 1996) combines SMART and AHP for multi-criteria decision analysis with uncertainty analysis. It supports trade-off analysis, budget prioritisation, project brainstorming and custom report generation. It also provides flexible assessment approaches either numerically or verbally. Many case examples are offered to demonstrate the functions of the system.

(13) DPL (Decision Programming Language)

DPL (Decision Programming Language) (Applied Decision Analysis, 1996) combines decision trees, influence diagrams and spreadsheets into the system to define decision problems. Multiple forms of outputs from the system are available including policy trees and summaries, distribution graphs, rainbow diagrams and tornado diagrams. There are also several problem examples to help the user understand the basic concepts of the package. A special model description language is provided in the advanced version of DPL to allow users to write DPL programs to define decision problems.

(14) CADET (Computer Aided Decision Evaluation Tools)

CADET (Computer Aided Decision Evaluation Tools) (AT&T Bell Lab., 1997) uses influence diagrams as a tool to frame decision problems in resource allocation for project funding. CADET has time series and evidence propagation capabilities. By providing tools for problem framing, sensitivity analysis (probabilistic and deterministic), and evidence propagation, CADET facilitates the processes of decision making and generates the action plan.

(15) DATA (Decision Analysis by TreeAge)

DATA (Decision Analysis by TreeAge) (TreeAge Software Inc, 1997) builds models as standard-notation influence diagrams or as decision trees. Influence diagrams can be automatically converted into fully configured asymmetrical decision trees. Markov processes, Monte Carlo simulation, cost-effectiveness analysis and multi-attribute modelling are available in the system. Many case examples are offered to demonstrate the functions of the system.

(16) DecisionTool Suite

DecisionTool Suite (Palisade Co., 1997) is a set of risk and decision analysis software, including @Risk, BestFit, PrecisionTree, TopRank, and Riskview. @Risk, based on Monte Carlo simulation, is a risk analysis add-in for Microsoft Excel and Lotus 1-2-3. BestFit is a distribution fitting solution finding the statistical distribution that best fits any data set up to 30,000 points. TopRank is a What-If add-in for MicroSoft Excel and Lotus 1-2-3. It can determine which values affect the bottom line the most and ranks them in order of importance. Riskview is for statistical distribution previewing. PrecisionTree is a decision analysis add-in for Microsoft Excel for structuring decision problems by using decision trees and influence diagrams. It allows users to create decision trees and influence diagrams in the existing spreadsheets. Decision trees are used to model the sequence of events in the decision problem while influence diagrams are to demonstrate the relationships between problem elements including uncertainty.

2.6.5 DSS Evaluation Principles

DSS evaluation is an activity that measures qualitative benefits and the output of a system by some criteria or principles, and it can be considered a fundamental step to implement more effective systems. DSS evaluation can check the appropriateness of the key elements of DSSs, including the processes of decision making, the outlook of the system interfaces, and the ultimate usability of the system. For a specific DSS, system evaluation offers a way to measure the decision making success and the continued use of the system. System evaluation is also critical in selecting a suitable DSS generator to implement a specific system. For a class of DSSs, it provides a guide for the future development of systems of the same kind. Ex-ante DSS evaluation, in which evaluation proceeds prior to development and/or implementation of a system, can estimate relationships between DSS characteristics and task set performance (Gardner, Marsden and Pingry, 1993). Knowledge needed to achieve optimality in DSS design and selection may be in the form of general principles of DSS design, implementation and use, or in the form of information about the performance of a specific DSS on the tasks of interest. General DSS theories and principles or development information are demonstrated on specific systems. In short, DSS evaluation is a fundamental means to ensure the success of DSS development.

DSS evaluation generally includes two aspects, i.e. efficiency and effectiveness (Evans and Riha, 1989; O'Keefe, 1989). System efficiency is objective and easily measured, such as time, money, the number of reports or the lines of data processed and printed. Efficiency for a DSS focuses on the speed of decisions or the cost of the decision making process. In contrast, system effectiveness is subjective since it is concerned with using the correct information and procedures in arriving at a decision and whether the system is beneficial to the people involved. Subjective criteria for effectiveness evaluation are very complicated in that it is difficult to verify success, that goals are sometimes immeasurable and that the environment evolves and changes over time.

One of the key differences between DSSs and other information systems is its emphasis on system effectiveness rather than efficiency (Evans and Riha, 1989). Efficiency of a system is not a major concern as long as the system can be effective. The DSS community has frequently emphasised increased decision making effectiveness as a system evaluation criterion. The evaluation of a DSS can be equated with the measurement of effectiveness (O'Keefe, 1989). In effect, system effectiveness is equated with system performance in the area of DSSs.

No unified and comprehensive methodology for DSS evaluation or validation has been suggested. Little is known about available effective and operationalised methods to evaluate or validate such computer based systems (Le Blanc and Jelassi, 1989; Borenstein, 1998; Finlay and Wilson, 1997, 2000). However, some examples of system evaluation methods can be found in the literature. For example, O'Keefe (1989) presents a generalised multicriteria method, which fits the nine guidelines reviewed for measuring system effectiveness and consists of seven stages and some formulae. These stages and formulae address objectives by way of indicators, and evaluate the present state of decision making, the expectancy of the situation after the introduction of the decision aiding system, and the performance of the system after implementation.

The term of validation is used by some researchers (Finlay, 1989; Borenstein, 1998) in the place of evaluation even though there are some differences between them. System validation is defined by Finlay (1989) as the process of testing the agreements between the behaviour of a system and that of the real world system being modelled, while the focus of system evaluation is on the software and its effect on the real world. However, there are many aspects that overlap between validation and evaluation especially in terms of qualitative performances of a system.

Criteria that are used to evaluate system effectiveness for DSSs are paid special attention in the present study due to the fact that these criteria are one of the considerations when modelling DSSs. Some useful guidelines for DSS features, which can be used as the evaluation criteria, have been identified in the literature (Ariav and Ginzberg, 1985; Meador and Mezger, 1984; Reimann, 1985; Reimann and Waren, 1985; Sussman, 1984; Waren and Reimann, 1985; Applegate, Chen, Konsynski and Nunamaker, 1987; Evans and Riha, 1989; Le Blanc and Jelassi, 1989; Adams, Courtney and Kasper, 1990; Buede, 1992; Gardner, Marsden and Pingry, 1993; Rizzoli and Young, 1997; Borenstein, 1998).

Applegate, Chen, Konsynski and Nunamaker (1987) suggest that the requirements for supporting decision making include: 1) access to a wide range of ad hoc data; 2) flexible access to both quantitative and qualitative decision models; 3) a flexible knowledge management system to capture decision knowledge and to enable linkages among interrelated pieces of decision information; and 4) the capability to support both group and individual decision makers. Their suggestion is mainly from the system design point of view.

Le Blanc and Jelassi (1989) discuss some criteria for designing and selecting DSSs, which fall into four categories (technical requirements, functional requirements, documentation and training, and vendor information) and were expanded for each category. The primary technical requirement for the computing environment is IBM and DOS compatibility. Functional criteria include database management, statistical analysis, spreadsheet, word processing, graphics creating, and file handling and exchange. Documentation requirements consist of external documentation and on-line help in context. The vendor criterion is basically about the reputation of the vendor.

Evans and Riha (1989) mention some criteria for GDSS consideration, including quality interaction, stimulation, suppression of individual creativity in a group context, and transfer of information from one individual to another in a group. These are really social variables that bear a direct relation on the quality of the decisions and thus on overall system effectiveness.

Adams, Courtney and Kasper (1990) suggest that eight DSS support factors, e.g. decision confidence, system reliance, availability of analytical tools, generation of feasible alternative solutions, information search efficiency, decision making efficiency, understanding the problem domain and decision making performance, should be used to select and evaluate DSS generators.

Buede (1992) proposes some detailed features for the design of DSSs, and criteria to evaluate them using multi-attribute utility theory. The overall evaluation includes three aspects, i.e., system performance, system user friendliness, and system purchase cost. System performance is further broken into six decision tasks, including goal formation, information processing, problem structuring, analysis features, option selection, and project control. Sub-elements are defined for some of these tasks. User friendliness is evaluated based on some subjective judgements on the part of users. For the purpose of DSS software survey, Buede (1996, 1998) and Maxwell (2000) include a range of questions sent to vendors of decision analysis software, most of which is MCDM based. These questions are primarily based on the system features and evaluation criteria reviewed in Buede (1992), and show a promise to become de facto software standards for commercial DSSs.

The desirable features for an environmental DSSs are identified by Rizzoli and Young (1997) from the position of artificial intelligence and experts systems, which deal with human knowledge representation and engineering in computation. The features include the ability to acquire, represent and structure the knowledge in the domain under study; the ability of the knowledge base (or domain base) to separate data from models for model re-usability and prototyping; the ability to deal with spatial data (for example, a geographical information system component); the ability to provide expert knowledge specific to the domain of interest; the ability to be used effectively for diagnosis, planning, management and optimisation; and the ability to assist the user during problem formulation and selecting the solution methods.

The major principles already discussed in the literature dedicated to designing and evaluating MCDA (Multicriteria Decision Aids) are integrated in the present study into a single frame, which contains two aspects, i.e. system performance and technical implementation, as shown in Table 2.1a and 2.1b. System performance deals with the functional aspects of decision analysis for DSSs. Technical implementation is concerned with the technical expertise for implementing DSSs using specific skills and tools. However, not every single DSS characteristic reviewed in the literature is included. Some DSS features, such as IBM compatibility, are outdated. Some, such as interfaces to particular applications, are not generally desirable to DSSs. Some of them, such as goal formation and problem understanding, are merged into one. Only those features that are essential and important are considered. These critical DSS features are used as the criteria to evaluate DSSs and also as a guide for further DSS modelling in this study.

Table 2.1a: DSS Evaluation Principles (DSS Performance)

DSS Performance
(1) Group Decision Making Support
(2) Guidance in Decision Making Processes
(3) Elicitation Techniques
(4) Problem Analysis and Structuring
(a) Problem Understanding
(b) Brainstorming Techniques
(c) Uncertainty Understanding and Handling
(d) Alternative Creation and Value Structuring
(e) Relationship Guidance
(5) Evaluation and Choice
(a) Results of Analytical Results
(b) Sensitivity Analysis
(6) Model building
(7) Decision Implementation Structuring
(8) Reporting

Table 2.1b: DSS Evaluation Principles (Technical Implementation)

Technical Implementation
(1) User Interface
(a) Command Interface
(b) Cursor and Screen Control
(c) Visualisation and Multimedia Support
(d) Tolerance to User Errors and System Responses
(2) Learning and Ease of Use
(a) Ease of Orientation
(b) Obvious Operations
(c) Demonstration Examples
(3) System Help
(a) Automated Tutorial
(b) On-line Help
(c) Text Manual
(d) Command Glossary
(e) Methodology Explanation
(f) System Status Displaying
(4) Data Processing
(a) Data Management
(b) Data Input and Output
(c) Data Interfaces
(d) No Constraints of Problem size
(5) Installation

Table 2.1a lists the items of DSS performance evaluation principles. Firstly, for MCDM decision problems, group decision support becomes essential since these problems involve multiple interest parties possibly from geographically dispersed areas. Tools and processes that have been devised for individual managers acting alone are unlikely to provide the full range of decision making support (Finlay, 1994).

Secondly, intelligent guidance is needed in all the phases of problem analysis, problem structuring and the final evaluation when making a decision by using a DSS. A user can easily be aware of what is being done by using a tool or method provided by the system, and how and what to do next.

Thirdly, versatile elicitation techniques should be available for users to elicit values for different decision elements during the phase of problem analysis and structuring, and to support elicitation for scores, weights, probabilities, decision alternative preferences, or risk preference during the phase of decision evaluation and choice.

Fourthly, in the phase of problem analysis and structuring, various techniques, such as brainstorming, decision trees and influence diagrams, are used to facilitate users to understand the decision problem under consideration and to identify and structure the decision problem. There are five sub-principles for the evaluation of problem analysis and structuring. Besides the support of problem understanding and brainstorming, there is a need for uncertainty understanding and handling, decision alternative creation and value structuring, and guidance of relations among objectives (decision criteria).

The fifth item of DSS performance evaluation principles involves the evaluation of decision alternatives and decision criteria, and the choice of a final result. After the elicitation of various values, judgements, and preferences, a DSS should offer effective, efficient and flexible analytical abilities in order for the decision participants to make the best satisfying choice or to obtain recommendations. This item includes two sub-principles, i.e. sensitivity analysis and result presentation of analytical results. The elicitation of analytical data is also an important aspect to be considered at this phase of decision analysis, and it is included in the third evaluation item above. As to the final analytical results, according to Roy (1973), there are three the fundamental goals of the decision maker, i.e. choice, sorting and ranking. The goal of choice allows the users to select the best out of a set of alternatives. The goal of sorting places decision alternatives into ordered categories. The goal of ranking presents a complete ordering of decision alternatives. These three goals may be considered for presenting analytical results in DSSs. After all, decision participants must assess the sensitivity of outputs to imprecision in the values of some relevant parameters.

The sixth item of DSS performance principles is about model building in DSSs. Decision models are problem statements that represent decision problems in an abstracted way, usually in a mathematical form. Decision model management in a DSS deals with decision model building and other model manipulations. For DSSs that use MCDM as the basic decision model, the task of basic model building in the system level for the decision problem as a whole may equate to the activity of problem structuring as MCDM problem structuring defines the decision elements required by the MCDM decision model. However, there may be some lower-level decision models needed for the analysis of specific aspects of the decision problem.

The seventh item of DSS performance evaluation principles is the structuring of decision implementation. Decision implementation need to be planned and structured before a schedule of tasks is completed to achieve the desired change from the status quo.

Finally, reports on the analysis processes and results are very important for decision participants and other interested parties to understand the decision being made. The user inputs, especially the problem structure, should be documented in order to justify the conclusions. The final conclusions of the analysis and other comments may also be reported through printouts and screen displays.

Table 2.1b lists the evaluation items of the DSS technical implementation, which mainly contains user friendliness, data processing, and system installation. Even though the technical implementation of DSSs is important and even critical sometimes, it is not considered as an essential part of system performance in terms of direct decision support.

User friendliness is of critical importance because ease of use and user acceptance are significant determinants of intention to use a computer technology. Users are not likely to adopt a system unless they perceive it as a useful and easy-to-use tool (Davis, 1989; Moore and Benbasat, 1991). There are three main aspects of user friendliness, namely, user interface, system learning and use ease, and system help offered for system operations.

A DSS should offer a friendly man-machine interface to enable the users to take advantage of it with ease. Several types of command interfaces, such as hierarchical menu, pull down menu, command driven interface and tool bars may be provided by a system. Positive controls of cursor and screen display should be available for a DSS to allow full screen editing and modifications of screen displays so as to tailor the information displays. Visualisation of decision analysis and the multimedia decision support are relatively novel ideas, but they are receiving attention in the implementation of DSSs (Dyer, Fishburn, Steuer and Wallenius, 1992). Visual MCDM is concerned with how to graphically represent multiple criteria decision problems.

Multimedia is a form of interactive computer representation of information by using a variety of media, such as text, graphs, voice, music, still image, live video image, etc, to create a finished presentation. For a friendly user-interface, the tolerance to user errors and quick system responses to data processing and input error diagnoses are also important features.

A DSS should be very easy for users to learn and use to carry out decision analysis. Although ease of learning and use of a system is perceived subjectively and psychologically (Davis, 1989), there are some issues that need to be examined for the purpose of system learning and use ease. Firstly, the users of a system should take little effort to learn about the system, i.e., what does each command (menus and commands input in the command line) mean, and how to use them. Secondly, the system should be designed so that most operations are obvious to common users. Finally, the system should be demonstrated by using some build-in examples in the system. Users can learn to use the system by using and modifying the existing examples.

In a DSS, help should be offered when necessary at any time during the system operations. Automated tutorials and on-line help have been recognised as significant and essential in the design of MCDA (Belton and Elder, 1994; Fernandez, 1996). A user's manual may be able to answer both start-up and profound questions. A detailed glossary of commands as well as their formats and functions can be very helpful as well. The explanation of the methodology adopted by the system to support decision making may make the users understand what happens behind the scene. System status information, such as memory available, status of data input, and numbers of criteria and alternatives, can inform the users of the status of the system and that of the decision making procedure.

Various data, such as values of variables, user judgements, problem elements, etc, are stored and processed in a DSS to provide necessary input to decision making activities and analytical results to users. The main aspects of data processing include data management, data input and output, data interfaces, and constraints to the size of the decision problem being solved. A range of DBMS (Data Base Management Systems) is available for consideration when implementing databases for a DSS. A system should support most data output devices such as colour monitors and printers, and should also support a variety of data input devices. Spreadsheets are attractive to DSS developers due to the facilitation offered to data entry and review by this kind of data input and output format. In a DSS, there should be data interfaces for common applications, such as spreadsheet-based programs (e.g., Excel and Lotus 1-2-3) and DBMS (e.g. FoxPro and Oracle). Interfaces to particular applications, such as another decision analytical tool and a geographical information system, are also desirable sometimes. There should not be any constraints to the number of alternatives, the number of levels in a value or decision tree, or the number of states of a node in such a tree.

The installation of a DSS should be easily achieved by normal users and will be automatically completed by the system itself. Equipment requirements of computer hardware and software should be reasonable compared to their utilities.

Although some other aspects, such as vendor information of a commercial DSS, system cost, organisational hardware and software environment, and human factors, may be considered when evaluating a DSS, only main considerations are listed in this study. Emphasis is placed on system performance criteria and technical implementation aspects.

In fact, the evaluation principles listed in Table 2.1a and 2.1b are served as guidelines for the modelling of DSSs for natural resource management in this study. The evaluation criteria for system performance are generic system features to support decision analysis. They are about what needs to be done for a DSS in terms of decision making tasks and requirements, and they can be used in modelling DSSs. The criteria for technical implementation are about how the system performance is met by dealing with detailed system implementation skills. They may be considered in the low-level design of a specific system and also in the actual implementation (coding).

2.6.6 Issues and Perspective

There are advances in the current DSSs. Most systems provide fascinating colour and graphical windows-based interfaces. Some packages include easy-to-use spreadsheets to facilitate data entry and review. Decision analytical software has historically focused on developing functionality that supports computing and sensitivity analysis within a given model structure. The current set of systems affords the analyst with plenty of choices for selecting a very powerful decision analysis tool (Buede, 1998). The recent years have also witnessed significant improvements in the model structuring and quantitative elicitation (Maxwell, 2000).

There is, however, some room for improvement in the current DSSs in terms of technical implementation. Even though basic concepts of visualisation of decision problem, criteria and alternatives, such as graphical representations, have been fulfilled in many systems, further exploration of visualisation is desirable for the entire process and the whole problem. All of the current packages in effect continue to make significant headway in improving the ability to visualise model results and sensitivities (Maxwell, 2000).

Efficient on-line tutorials, especially those with animation, have not been implemented widely (Decision Explorer by Banxia Software Ltd offers the most attractive tutorial, but has only animated text contained in rectangles). As a result of continuous efforts of current software market, advanced and effective computer techniques, such as animation, multimedia, and speech recognition (in the far future), would play an important role in the DSS development in the future.

Bhargava, Sridhar and Herrick (1999) make an interesting observation about the user interfaces of the 11 commercially available DSSs that most products provide general-purpose inflexible user interfaces. These systems do not distinguish between the user interface for analysts and that for common users or decision makers. As a result, all users have to interact with the systems via the same interface as that for the analysts.

Besides, the primary focus of the current DSSs continues to be on analysis features rather than on support for non-analysts in structuring the problem and a more complete set of elicitation support (Buede, 1998). Although these analysis features are useful to all users and critical for the insight that decision makers need, there are relatively few of us that are qualified and comfortable in using these packages. It is claimed that these packages will never become as widespread as spreadsheet and database packages as long as the user has to be a highly educated analyst.

In terms of methodological research in the development of DSSs, there is substantial room for improvement, especially in the support for the problem definition and for problem structuring (Buede, 1998; Maxwell, 2000). The function of initial problem structuring in DSSs is critical in decision analysis though analysis facilities are important. Hersh (1999) discusses issues arising from the planning and management of water resources and power generation and examines examples of the use of DSSs in the literature in these areas of sustainable decision problems. Hersh (1999) also suggests that there is a need for the development of DSSs which facilitate structuring problems and defining goals.

As a fast growing segment of the software market, according to Buede (1998) and Maxwell (2000), group decision support is emerging as a development focus to allow analysts to work with multiple stakeholders in collective groups and distributed locations. As a result, a new emphasis is paid to web-based decision support. The World Wide Web, which is also known as the Internet, is increasingly being used as the platform of many people due to its network and platform-independence and very low installation/maintenance costs. The Internet and corporate internets (internal networks in organisations) have opened a wide possibility of building DSSs to deal with problems of a global nature. Web-HIPRE (Helsinki University of Technology, 1998) is an example of software available for use on the Web.

Eom, Lee, Kim and Somarajan (1998) suggest that the urgent challenge in the field of DSSs is that of bridging the gap between practitioners and DSS researchers. The conclusion results from the study of surveys of DSS applications during 1971 to 1994 (Eom and Lee, 1990; Eom, Lee, Kim and Somarajan, 1998). Research on DSSs has mainly concentrated on DSS components, for example, data, models, and dialogues. Future DSS research should redirect its attention to underdeveloped subspecialties to provide useful guiding principles for practitioners in the integrated processes of design, implementation, and evaluation of systems. The DSS theories need to be developed to deal with these aspects of DSSs for practitioners, and put these theories and their related concepts, frameworks, techniques and tools in practice. Murphy (Schneymann, Graves and Murphy, 1991) is quoted to demonstrate the situation in developing DSS theories for practice, and says that development depends on "maintaining a constructive tension between the immediate needs of managers and the research interests of professors".

Maxwell (2000) noticed that there is a need for DSSs that support specialised application of decision analysis to a particular class of problems. Most systems surveyed (Maxwell, 2000) are designed to support generalised applications of decision analysis to common decision problems. Systems specifically designed for a problem domain, such as health care, finance, transportation and natural resources, are especially desirable and useful to the practitioners of decision analysis in that field.

Halsall and Price (1999) suggest that DSSs are not affordable to most small enterprises in the manufacturing industry. A prototype DSS, which is based on the modelling of the manufacturing processes, is developed to demonstrate the way to cope with this situation. Static and dynamic data about the manufacturing system and production planning information is stored as a result of the modelling.

Bhargava, Sridhar and Herrick (1999) show that complexity and long development time are inherent in building DSSs, resulting in preventing the wide use of them. Building a DSS requires some significant expertise such as decision analysis and programming. The task is further complicated by the fact that a DSS may also need to connect in real time with other enterprise applications. These reasons are claimed to be the deterrents to the wide use of DSSs. A DSS generator is suggested as a current solution for developing an application specific DSS. Using a DSS generator reduces DSS development to a decision analysis task, which requires expertise in decision analysis and mathematical modelling rather than programming skills. Bhargava, Sridhar and Herrick (1999), however, do not pay much attention to the difficulties for non-analysts to model the decision problem with expertise from decision analysis in such a specific DSS.

In natural resource management, the issue of complexity and difficulty of DSS development and utilisation for practical decision problems may be worse than for general DSSs since natural resource problems are complex and multidisciplinary in nature. Natural DSSs are complex applications (Batachia, 1999), possibly integrating different advanced technologies and requiring high research and development efforts. DSSs for natural resource management are time consuming and costly to develop and maintain. As a result, these DSSs are difficult to adapt to rapidly changing decision environments. A sound DSS generator in the field of natural resource management is an invaluable asset to the practitioners of decision analysis and also to stakeholders in a decision problem.

Generally, the DSS literature has not paid sufficient attention to the use of evolving or meta-methodology in the development of DSSs. Moore and Chang (1983) suggest that DSS designers should be aware of certain issues above and beyond the specific details of any particular design engagement. The idea of meta-design is proposed to enable individual DSS designers to develop their own design frameworks appropriate to their particular needs.

In the area of GDSS, despite the increasing number of published studies, little progress has been made toward theoretical models that integrate the existing empirical observations and that offer guidelines in developing effective group systems (Rao and Jarvenpaa, 1991). The motivation for the theoretical models stems from two interrelated sources: the need to reconcile the inconsistent results across studies and an assumption that theory-based research will advance the area. The inconsistencies in empirical results are argued to be at least partly due to the lack of well-articulated theoretical models for developing hypotheses and interpreting results. A theoretical model can provide the commonality necessary amongst researchers to build a clear and rapid understanding of GDSS.

Wang (1995) notices that there is no tool commonly used for the domain analysis and modelling of DSSs despite the vast amount of literature on the research of DSS theory and applications. Little research has been forthcoming with regard to analysis techniques for DSSs. The objective of using a DSS domain analysis technique is to determine the requirements for resources in supporting certain types of decisions as well as the possible paths involved in such decisions. The outcome of a DSS model should be of assistance in the design and implementation of the DSS.

Wang (1995) suggests that the major reason for the lack of DSS modelling techniques can be traced to the inadequate ability of traditional techniques to model DSSs. Traditional techniques of information systems analysis, such as the data flow diagram (DFD) for modelling data flows, have focused on system functions, data processing and input-process-output transformations, but have their limitations in modelling DSSs. In the DSS context, input-process-output transformation is no longer the major characteristic of DSSs. The DSS analysis must involve decision strategy analysis, data analysis, and technical analysis. In DSSs, user-computer interaction and timing are representative of the dynamic aspect of a system. Traditional methods do not support these kinds of system modelling.

It seems that there is a great need for domain analysis and modelling for DSSs with an effective methodology to assist the development of DSSs for natural resource management as well as determining the resources and paths for decision making in natural resource management. This methodology can model these DSSs in a scientific way and reduces the requirements for artistic skills in modelling DSSs. A DSS for a specific decision problem may be specified as a sketch of the DSS model resulting from the modelling. The DSS model can deal with the complexity of DSS development and make the development of a specific DSS affordable in terms of both time and cost. Ideally this methodology should be integrated with that for modelling decision problems since a uniform methodology and philosophy is very desirable in decision analysis and DSS development to allow the improvement of effectiveness and efficiency in both aspects (This is further discussed in Chapter 3). Relevant entities, either tangible or not, such as problem features, decision properties and system functions, in decision analysis and DSSs, may be modelled in a simple and transparent way. Different users, including analysts and non-analysts, are represented in the DSS model and in different ways, allowing them to interact with DSSs in a comfortable and qualified way. The DSS model is the mechanism to bridge the gap between the DSS researchers and decision analysis practitioners as it renders the DSS design, implementation and evaluation understandable to decision practitioners and DSS researcher alike. Finally, an ideal DSS model can support the development of network-based GDSS for all the phases of decision making, including problem structuring.

The potential of object orientation is explored in the application to model DSSs for natural resource management with MCDM. The next section reviews the application of object orientation in the modelling and development of DSSs.

2.7 Object Orientation in the Modelling and Development of DSSs

The development of DSSs includes analysis, design and implementation. Analysis mainly concerns system requirements and theoretical frameworks. Design and implementation focus on system architectures and coding with a specific programming language under certain operating platforms. Modelling of DSSs aims at building system models for various purposes, mainly analysis and design of a specific DSS.

2.7.1 Development of DSSs

Dolk and Konsynski (1984) contend that the outward structure of the model representation in DSSs can be based on that of the object-oriented programming language in that it consists of data objects, procedures, and assertions all expressed in first predicate calculus. A predicate is that feature of language which can be used to make a statement about something, e.g. to attribute a property to that thing. The data object section enumerates the data items and types comprising the structure being described. A data type may be another abstraction. The procedure section lists each procedure, the data objects it accesses, and the data objects it returns. The assertion section specifies information about the data objects and procedures and their various relationships. Data items, data types, and procedures are assumed to be predicates while assertions are well-formed formulas in the predicate calculus. A simplified instance of a model representation for a generic linear programming (LP) model is given. An LP model expressed in equation form can be regarded as having five data objects: 1) an objective function (OF) to be optimised, 2) subject to one or more constraints (CON), 3) evaluated over parameters (PAR), 4) evaluated over index-sets (IDXSET), and 5) yielding values for decision variables (DV). Procedures in a model abstraction fall into several different categories of operations: 1) data object manipulation, 2) Boolean, 3) transformation, and 4) model solution. Assertions in a model abstraction are equivalent to rules governing the behaviour of the model being described.

Hagmann (1988) produces a Local Area Decision Network (LADN) model for Small Group Decision Support Systems (SGDSS) that integrates methods of organisational decision making, object orientation, fuzzy sets theory, and Petri nets. The model is defined in objects which encapsulate the activity of the data involved in the consensus seeking process of decision making. The DSS provide abstract data objects to provide a transparent environment for the user, fuzzy set theory techniques to evaluate and rate the competing alternatives and goals, and local area decision network for interaction between members and the system, and access to other applications of the organisation for the acquisition and use of information. The objects to implement the SGDSS are defined and Petri nets are used to characterise their interaction. It is stated that the SGDSS create an environment conducive to the improvement of the quality of decision making by small groups.

Le Claire (1989) proposes an object-oriented architecture for a DSS in a personal computing environment. A prototype DSS is implemented using the Smalltalk/V object-oriented programming system. Users develop their data and model representations following a diagrammatic technique. They are able to develop and implement classes of models such as the transportation model, a general linear programming model, a network model, and an assignment model. The DSS architecture permits users to follow a three step progression in problem solving from conceptual abstraction, to operationalisation, and finally to implementation.

Lal (1992) designs a farm machinery DSS, which combines object-oriented and knowledge-based simulation, an expert analysis system, and an intelligent information manager in a logic language, namely, PROLOG. The system successfully simulates field operations for a complete cropping season, and responds correctly to scheduled work hours and to the withdrawal of machinery and labourers. PROLOG facilitates simulating field operations in an object-oriented manner. Expert systems and other components are incorporated with the inference capabilities of PROLOG. It is claimed that the descriptive nature of PROLOG allows the modelling of aspects of the DSS that are typically ignored or difficult to model when using conventional approaches.

Dolk and Kotteman (1993) note that numerous authors have recognised the applicability of the object-oriented phenomenon to model management in DSSs and the associated benefits of models as objects, solvers bound to these models, and inheritance hierarchies. According to Dolk and Kotteman, this has probably done as much harm as good in the advancement of model management research. All roads lead to object-oriented environments for the implementation of integrated modelling environments. This is not surprising since models are complex data structures requiring complex manipulations. However, it is stated that object-orientation is not a substitute for a theory which encompasses model manipulation as well as representation, and that object orientation is primarily an implementation choice for building modelling environments rather than a substitute for model theory. Dolk and Kotteman (1993), however, do not discuss in detail why object orientation cannot be used for representation. However, other researchers (Muhanna, 1993; Lenard, 1993; Pillutla and Nag, 1996) may not agree. Pillutla and Nag (1996) develop a schema to represent model information in an object-oriented framework that relies on the definitions of natural entities. Lenard (1993) uses a relational database management system to represent model objects and to implement the object-oriented model management system.

Muhanna (1993) proposes an object-oriented framework which provides a unified context for model management and DSS development. The proposed object-oriented framework synergistically integrates two proposals, namely Geoffrion's (1987) structured modelling, which provides a good methodology for modelling-in-the-small, and a system framework which furnishes a number of concepts and structuring principles which are fundamental for capturing the semantics in a modelling environment and addressing issues related to modelling-in-the-large. The two proposals are cast in terms of object-oriented concepts. The framework permits us to uniformly treat entities in the environment as objects. These entities could be users, concepts, models, subroutines, and even windows in a user interface. It is argued that an object-oriented approach can significantly contribute towards coherently integrating the myriad proposals in model management in the literature. It is also contended that the object-oriented approach's major contribution to model management and DSS development lies in the fact that it reflects a natural view of the world modelled in databases, software and analytical models. No methods for actual DSS development are proposed in the paper.

Waxlax (1993) uses a context-sensitive problem solving system, STRATEX by Nokia Corporation, to build an "object-oriented" DSS to support managers by explicitly linking strategic visions and objectives to established management processes. The developed system is claimed to be very useful for strategic management. However, it is unclear about the grounds on which the system is called an "object-oriented DSS".

Du (1995) presents a hybrid methodology, which integrates object-oriented databases, fuzzy logic controllers, neural networks, and active systems, for the development of a DSS. The knowledge base (the fuzzy logic controllers and neural networks) can be integrated with the object-oriented database so that the data can be organised statically and the system can be operated dynamically. The application of object orientation is only limited to the administration of the database in the system.

Bomme and Zimmermann (1995) describe the architecture of an intelligent symbolic object, which is a system defined in relation to a database and a rule based expert system, for decision making procedures based on the object-oriented approach. Activation of rules attached to the objects invokes intelligent behaviour in a decision making procedure. The symbolic object is then compared to the objects in the database in order to generate a classification based on customised schemes for decision making activities.

Rafanelli, Ferri, Maceratini and Sindoni (1995) report a DSS in health resource allocation. An object-oriented database is used to store scenarios depending on goals and constraints in a decision problem. The system also contains a geographical information system and some problem solution algorithms to allocate new resources to optimise the cost/benefit ratio.

Fedra and Jamieson (1996) describe the WaterWare system, an information and decision support system for river basin management. The basic framework of the system combines a hybrid geographical information system as the overall structure with classes of objects in object orientation, including river basin elements, models and model scenarios, tasks or decision problems. The states of river basin elements are determined by a set of methods, which are models or sets of rules for an embedded expert system. Tasks are specific problem oriented views of river basin objects, and represent their states to the user to support planning or management decisions. The various objects are linked explicitly. Models such as an irrigation water demand estimation model are used to update the states of these respective objects, and thus provide inputs to a water resource model.

Chen and Sinha (1996) discuss an object-oriented approach for the construction of an integrated DSS. High-level constructs of models and data are system-tier objects and are used to develop the object-oriented framework. An analysis of computational needs at various levels defines the data objects of the system. The inheritance relation of the object orientation provides the integration of the multi-level decision making. Implementation-tier objects of a system are descendants of the system-tier objects. An inventory management system is implemented to illustrate the use of the DSS (Chen, 1996).

Missikoff (1998) describes the analysis, design, and fast prototyping of an information and decision support system for railway traffic control with object orientation. The system tightly integrates information management and problem-solving functionality by means of an object-oriented approach. A knowledge-based approach is used to model the railway traffic control problem. In the architecture of the system, particular attention is paid to the database component and train conflict-solving capabilities.

Rizzoli, Davis and Abel (1998) present an object-oriented system architecture for the management of environmental models in DSSs. The object-oriented approach emphasises the separation of models from data, thereby promoting model and data integration and re-use. It is demonstrated that a DSS can use this approach to implement the model management to facilitate problem definition and problem solution.

2.7.2 Modelling of DSSs

Holtzman (1989) envisions the architecture of intelligent decision systems based on object-oriented programming, which would be likely to group procedures and data structures into comprehensive objects, and which takes a quite different form from that of traditional systems.

Muhanna (1993) suggests an overall object-oriented approach, in which each method, tool, modelling technique, and software engineering activity is either object-oriented or supportive of an object-oriented approach, for model management and DSS development. More specifically, the object-oriented approach provides a framework for analytical modelling, database modelling, solver implementation, and general application development. The benefits of the object-oriented approach would be enhanced if it is applied consistently throughout since besides the conceptual integrity it affords, this consistency could offer the value-added benefit of simplifying integration both within and across the following categories: models, solvers, databases and various support utilities.

Wang (1995) makes an attempt to add the object-oriented approach in DSS domain analysis and modelling. An object-oriented DSS domain analysis and modelling technique based on the Coad/Yourdon method is proposed. Five fundamental types of object classes are identified to support all the aspects of DSSs by using the object-oriented paradigm. These identified classes model DSS components in an object-oriented way. A list of object classes was extracted from the general literature of DSSs. This is done through the identification of nouns in the descriptions of DSS structures and uses, as well as operations. Next, a shorter list is abstracted by checking across the object classes identified in the first stage. The final set of object classes of DSSs derived includes: 1) Physiomorphic entity- a physically existing entity (all entities targeted by traditional relational database, such as environmental entities, physical resources, and organisations, and in a broader sense, for example, hypertexts and time series); 2) Instrument - an elementary model (elementary models, such as a statistical model, mathematical programming, cognitive maps, spreadsheet); 3) Software environment - computer software which the decision maker can access to interact with the system; 4) Decision solver - a decision path or procedure; and 5) Evidence - an output from the DSS.

The five classes identified by Wang (1995) may be useful in helping the decision analyst understand the decision making facilities and use these facilities to facilitate decision making activities. They, however, cannot contribute much to the actual implementation of a DSS. In addition, the five fundamental object classes may or may not be sufficient for complicated DSSs because, according to Wang (1995), the identification of the fundamental object classes in the study is based on a survey of a relatively small DSS sample.

Gauthier and Neel (1996) build an object-oriented decision support framework for knowledge management and decision support in the area of agro-ecosystem management. The object-oriented paradigm is believed to provide a foundation for the construction of a general multifaceted and comprehensive decision support framework enabling the integration and use of different types of knowledge and information processing tools. A library of Smalltalk classes constitutes the framework onto which developers can build systems to represent agro-ecosystems and to support the management of these systems.

To summarise, due to its success in the development of general software systems, object orientation has been involved in some aspects of DSS implementation, such as programming, architectural design, model management and user interface development. Object-oriented programming languages are being used in the implementation of most DSSs. Object-oriented architecture design provides a detailed structure of system construction for DSSs that are typically implemented in object-oriented languages. Model management is carried out in an object-oriented way as decision models are implemented as objects in object orientation. User interface is being developed with reusable components from user interface class libraries, which store all reusable user interface components.

On the other hand, however, the literature is very limited on techniques for object-oriented DSS analysis and for object-oriented support of the DSS development life-cycle (including analysis, design, and implementation), although most developers of DSSs are using object-oriented programming languages to implement DSSs. Most importantly, very few object-oriented methods have been found in the modelling of DSSs to support DSS development in the literature. Little research has been conducted on the domain analysis and the object-oriented modelling of DSSs. This situation of research in object orientation applications in the modelling of DSSs is more or less similar to that in decision analysis and problem structuring. Object orientation at this stage mainly focuses on the analysis, design and implementation of traditional software systems.

In short, few comprehensive object-oriented methodological principles have been incorporated formally into the literature for decision analysis and DSSs. Nevertheless, object orientation has demonstrated a promising future as a pragmatic methodology in these aspects (Booch, 1991; Muhanna, 1993; Graham 1994; Wang ,1995; Gauthier and Neel, 1996; Missikoff, 1998; etc).

2.8 Selection of Object Orientation Techniques

At present there are probably well over 50 more or less complete object-oriented methods in existence according to Graham (1994). Among the existing object-oriented methods, the oldest is probably due to Booch (1986). For more examples, see OOA/OOD (Object-Oriented Analysis and Design) (Coad and Yourdon, 1991a,b), SOMA (Semantic Object Modelling Approach) (Graham, 1994), OOAD (Object-Oriented Analysis and Design) (Martin and Odell, 1992; Martin, 1993), OMT (Object Modelling Technique) (Rumbaugh, Blaha, Premerlani, Eddy and Lorensen, 1991), OOADA (Object-Oriented Analysis and Design with Applications) (Booch, 1994), and Object-Oriented System Analysis (Shlaer and Mellor, 1988, 1992).

The area of object orientation, however, is not well bounded where research is still incomplete in certain aspects. Most of the techniques in object orientation are essentially concerned with developing software systems, and object-oriented analysis mainly focuses on the analysis phase that is after strategic planning. In system analysis, a more abstract approach is required at the initial stage, with details added progressively as analysis proceeds. Even in the area of design, there are still some issues to be solved. For example, current methods for object-oriented systems design do not adequately confront issues raised by the use of object-orientation in distributed environments (Purao, 1995). Designers (rather than analysts) of object-oriented distributed systems have no accepted guidelines on which they may base their distributed decisions. In the current object-oriented analysis methods, though some, such as Coad/Yourdan, are simple, they lack support for describing system dynamics. Some, such as Rumbaugh's OMT and Shlaer/Mellor, are rich in semantics but very complex to learn. Many methods such as OMT help little to express business rules and constraints, although some allow rules and constraints are added as an afterthought.

Nevertheless, the future of object-oriented methods looks very promising in that the object metaphor appears to be the most natural one to adopt in real practice. Object-oriented approaches have received attention in management information systems development due to the advantages over the traditional structured approaches. There is a widespread belief that object-oriented methods are better in many aspects than traditional approaches.

The problem of the method selection is not critical in the reality of object orientation as long as the main features of object orientation are maintained in a specific method. As Gossain suggests as a result of experiences (Gossain, 1998), the method is not even the major factor contributing to a project's success. Practitioners cannot simply pick up and follow a method. Organisations and problems vary too much for one off-the-shelf approach to be effective for all. The techniques that are used in the study for object-oriented analysis and modelling do not adhere to any specific method or a particular framework, but rather to explore the nature of object orientation in decision analysis and DSS domain analysis.

Hereafter, no specific practices of a certain method are stuck to in the present study. Some notations and prototypes of concepts are however borrowed from some object-oriented methods. The basic notations from UML (Unified Modelling Language) (Booch, Rumbaugh and Jacobson, 1999) are adopted and most of techniques from the

Booch method (Booch, 1986, 1991, 1994; Martin, 1996) are used for modelling the decision problems, decision making procedures, and DSSs for natural resource management. Both of them are revised in many aspects, such as the notational modification and addition, and with added facilities as can be seen in the following chapters, such as DSS evaluation principles guidance in system analysis, system use cases, actor-oriented object message diagrams and behaviour analysis.

The notations from the Unified Modelling Language (UML) are used in the modelling of the decision problem, decision making procedures, and the domain analysis of DSSs.

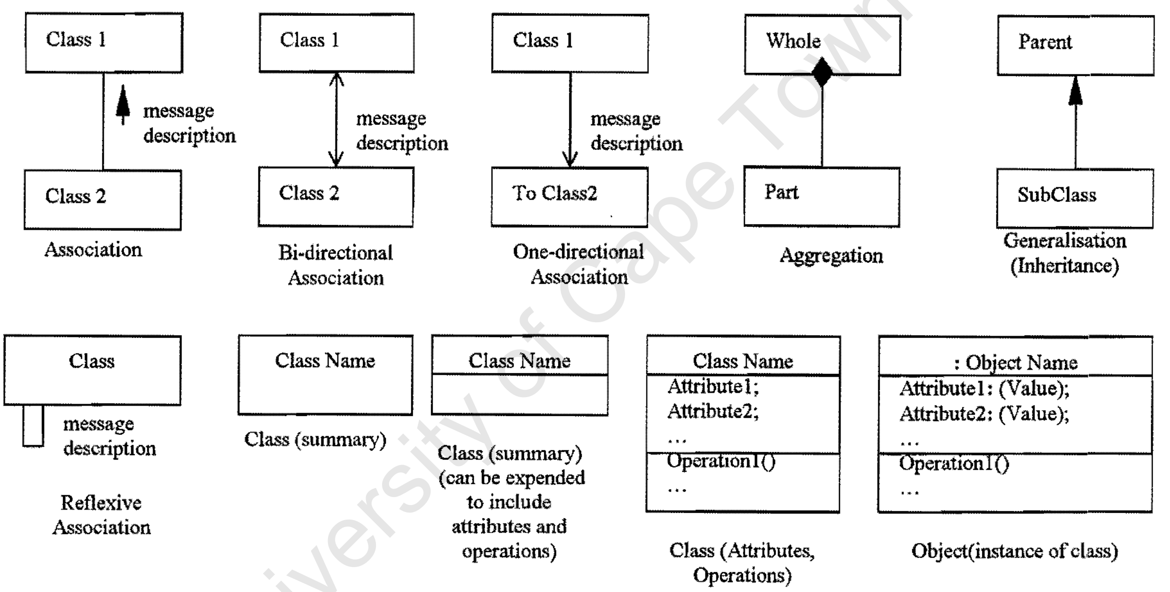


Figure 2.1: Basic Notations of Object Orientation

The diagrams in Figure 2.1 describe classes, objects, and their relationships. A class is a description of a set of objects that share the same attributes, operations, relationships, and semantics. An object is an instance of a class with concrete values and operations. Graphically a class and an object are rendered as a rectangle while a colon is put before the object name. An association shows a static relationship between two classes, indicated by a line connecting the two classes. It is a general relationship that is normally bi-directional but can be one-directional. Messages are sent between associated classes. A message is a specification of a communication among objects that conveys information with the expectation that activity will ensue. The receipt of a message instance may be considered an instance of an event. Most of associations can normally be navigated in both directions since an association describes two operations: sending and receiving. For example, in the association of "one person beats another" there is a "beat" operation in the "one person" object and a "hurt" operation in the "another" object. Only one function, however, is modelled in most of the object-oriented methods, depending on the assignment of responsibilities and functions of classes. Messages transferred between two classes are listed beside the connecting line between them. Messages may be passed between the objects of the same classes. Associations of this kind are referred to as reflexive associations, and are indicated as a self-pointing segments of line into the class itself.

A general association may be specialised as other relationships such as inheritance and aggregation. An inheritance relationship describes how a class is extended to satisfy specialised needs, represented by a solid-headed arrow pointing to the inherited class. An aggregation indicates how parts relate to the whole, represented by a filled diamond. A line with an arrow indicates a flow of control or message.

2.9 Conclusions

A literature survey of the main relevant research topics for this study was contained in this chapter. Included were some general aspects of the application of MCDM in natural resource management, decision making processes, problem structuring and DSSs. Special attention was paid to issues and findings related to problem analysis and structuring and DSSs. Problem analysis and structuring are the fundamental and decisive phases in decision analysis as they identify and define the decision problem. DSSs are the computer tools used to facilitate decision making. The application of object orientation in decision analysis and problem modelling, and in the development of DSSs was reviewed.

It was noted that a philosophically solid methodology in decision analysis and DSS development, should model the real world in a simple and natural way, be flexible to contain other modelling techniques, and be able to represent knowledge and experiences and use them in various aspects of decision analysis and DSS development. Such a methodology will uniformly model decision making and the domain of DSSs, bringing them under one roof. The object orientation base methodology might be able to achieve these goals.

However, little research has been reported in the literature about the comprehensive application of object orientation in decision analysis and DSS development. Very few object-oriented applications have been found in the modelling of decision problems, decision making procedures and DSSs in the literature.

Notations of object orientation were presented by the end of this chapter although no specific object-oriented method was chosen for the study. This was based on the consideration that the philosophy behind object orientation is most important in the application and that it makes no significant difference to select a certain method.

One of the objectives of this study has been to propose a philosophically and methodologically sound approach based on object orientation for the modelling of DSSs and decision analysis. Object orientation has proved promising in modelling the real world and its systems in a natural way. A general object-oriented methodological framework for problem structuring and DSS development is presented and discussed in the next Chapter.

Chapter 3

The Methodology for Problem Analysis and Structuring, and DSS Development

3.1 Introduction

Decision analysis has become one of the most active and interdisciplinary research fields in management science and operations research. MCDM has been a popular research area for more than two decades. Over the years many approaches and underlying theories have been developed for solving decision problems with multiple criteria. Solving decision problems via decision analysis can be divided into several steps: problem analysis, problem structuring, evaluation, choice, and implementation. Problem analysis addresses the importance of identifying the correct problem. Problem structuring, which is viewed as the core of the decision making process, is to find the decision elements, factors and their relationships. Evaluation concerns the elicitation of individual preference and judgement, and mathematical models for the problem solutions. Choice involves bargaining, consensus, and final selection.

Practitioners of decision analysis generally agree that problem analysis and structuring are the most important and difficult steps of the analysis. Indeed, the most critical and most time-consuming task in decision analysis is to clarify issues and relationships, and to identify quantitative and qualitative variables. Yet, until recently, most decision analytic research has all but ignored the initial steps, concentrating instead on questions of evaluation and choice. As a result, to some extent, problem analysis and structuring are still considered the "art" part (von Winterfeldt, 1980) of decision analysis. There is a need to turn this art into a science.

Decision analysis therefore tends to be of low productivity and an art instead of a science. It is also noted that DSS development tends to be a low productivity activity. The situation gets worse when it comes to complicated contexts such as natural resource management problems. As we know, even seasoned practitioners are repeatedly surprised by how much effort is needed to achieve useful results. According to (Geoffrion, 1987), there are four factors contributing to low productivity facing the management science/operations research (MS/OR) community. The first factor is that multiple problem representations are typically used in different situations. For communication with people such as managers and decision makers, a "natural" way of representation is used. For the analytical purpose, mathematical representation is generally used. A computer executable representation is still needed for the computerisation. The second factor, according to Geoffrion, is that the laborious task for the users to model the problem at hand in a format acceptable to the chosen solver requires specialised skills. The third factor is that most of the existing software only addresses one among the many models needed to solve a wide range of problems. The fourth factor is that most methods and software only cater to one or two steps of the problem solution. MS/OR practitioners are forced to combine different methods and software to deal with the entire procedure for a project. It is asserted that these factors contributing to the low productivity in MS/OR are also applicable in decision analysis. These four factors, however, are not the decisive ones resulting in the low productivity of decision analysis and DSS development. The two main factors resulting in low productivity are the nature of decision analysis as an art instead of a science, and the difficulties in reusing the past experiences and relevant knowledge when making a decision and developing a DSS. For decision analysis and DSS development, there is a need for a simple and transparent methodology that should be able to reuse the past experiences and relevant knowledge.

Object-oriented multiple criteria decision analysis is an attempt towards this end. Object orientation may contribute to the productivity of decision analysis and DSS design in two ways. First, object orientation will be able to utilise the research outcome from the literature and the experiences from the previous case studies by the mechanism of reusing. It may be argued that a decision problem is usually unique; that is to say, there will never be two identical cases. But, no one can deny human beings always make decisions based on knowledge and experience. The point is how to reuse our knowledge and our experience. Object orientation offers a mechanism, which is the inheritance of reusable objects, to reuse the existing proved knowledge and past experiences of a similar decision context. Secondly, object orientation provides a uniform tool to deal with almost all the aspects of decision making and DSS design.

Most of the current methods typically cater to just one or two of the many phases of the total life cycle of decision making. Users are forced to piece together a patchwork quilt of tools to deal with various phases as they arise over the life of a project. Object orientation will allow these phases to be carried out in a uniform and coherent way.

In this chapter, the methodology and the philosophy based on object orientation for decision analysis, especially in problem analysis and structuring, and DSS development are discussed. The remainder of this chapter is organised as follows. Section 3.2 provides further background discussion and motivation as well as the core object-oriented concepts. It discusses the benefits obtained from object orientation in comparison to the existing methods and argues that the methodology can meet the challenges of low productivity and the "art" problem in decision analysis. Section 3.3 suggests an overall diagram of the methodology for object-oriented decision problem analysis and structuring. Section 3.4 proposes the methodology for object-oriented DSS development. In Section 3.5, the two methodologies are naturally integrated under one framework for both decision analysis and DSS development. Conclusions are contained in Section 3.6.

3.2 Methodological Background

The general ideas behind the object-oriented decision analysis and development of DSSs include reuse of conceptual and technical results from previous analyses and designs, efficiency of decision analysis and DSS implementation, and assistance to understand the decision making and domain problems. A model for a specific decision problem can be obtained by instantiating the constructed object-oriented model for natural decision problems by specialising all the classes in the model to objects. This instantiated model may help understand the problems, get decision participants to have a basic knowledge of the decision making issues, and use the DSS generated with the same method.

Reuse is an important topic in software system development. Reuse of analysis, design, and implementation components is a powerful facility, which leads to dramatic increases in productivity and quality. Reuse can be applied to requirements, domain knowledge, design efforts, code generation and documentation (Biggerstaff and

Richter, 1987). In reusing existing information and products, costs are decreased and accuracy is increased.

The object-oriented framework provides a philosophy and a methodology which cater for both decision analysis and DSS implementation in a unified context. As shown in the object orientation, the real world, including the decision problem and the DSS, can be naturally modelled by a collection of objects, each of which can be represented in a simple and transparent way. Different viewpoints of the problem and the system can be observed from various hierarchies of objects.

The object-oriented framework brings about flexibility to decision analysis. The environment in decision analysis constantly faces new problems whose level of complexity keeps increasing. Due to its flexibility in handling changes and new problems and its ability to adapt to the particular problem on hand, the object-oriented approach is thus suitable for modelling such an environment. With object orientation, problem analysis can also start from individual objects with a bottom-up approach as well as from the overall system object with a top-down decomposition approach. Moreover, soft analysis methods can be applied in identifying the objects, which may include some hard aspects of problem features.

In object-oriented decision analysis, each party involved in the decision making processes is modelled as an object. Their communications and interactions are represented as messages transferred between the objects. This makes the methodology sound for applications to group decision making with geographically and culturally dispersed individuals.

The four main streams of problem structuring thoughts proposed by (Woolley and Pidd, 1981) are naturally met by the object-oriented decision analysis methodology. A step-by-step procedure for object-oriented problem structuring is described, mainly including actor identification, object identification, and object behaviour analysis. Objects of various kinds, such as decision elements and DSS components, and their relationships define the problem and the system. An understanding of the problem situation can be obtained through the models of problems, decision making procedures and the DSS architecture. Actors are identified and analysed at the very first stage of problem structuring and DSS analysis. Their perceptions to the real world are then modelled and are used to help understand the problem.

Classes are collective sets of objects with same features. Problems, problem elements and DSS components are classified as classes. This is a fundamental way to represent and make use of knowledge and experiences. Knowledge and knowledge representation in decision making actually have been used in every case of decision problems without exception. Decision makers use various kinds of knowledge to make a choice. Probability, for example, can be used to encode prior knowledge. A major concern in an intelligent decision systems is of the processing of knowledge, which, according to (Holtzman, 1989), contains at least five categories: domain knowledge, preference knowledge, probabilistic knowledge, user data (knowledge of the circumstances and information of the individual decision maker), and process knowledge. Each of them has important special features that can affect the way it is represented and plays in the system.

Actually knowledge can be coherently integrated with object orientation. In Artificial Intelligence (AI), a science of human knowledge representation and usage, rules can easily represent causal knowledge and rule bases are the basic form of artificial intelligence. Rules, however, can be represented in terms of attributes and behaviour in objects. Three types of knowledge for decision making, including preference knowledge, user data, process knowledge and probabilistic knowledge, may be represented in the object containing a series of attributes and methods (Graham, 1994) while domain knowledge can be represented as decision elements. In AI such objects are usually called frames though sometimes called units, or scripts for procedural abstractions. Frames of AI were invented to represent stereotypes of objects, concepts or situations, but their implementations are almost the same as the objects of object-oriented programming.

The analyst in decision analysis has always faced the problem of how to reduce the multifaceted knowledge in people's heads to a form that could meet the rigid tests of explicitness and consistency required by a computer. The object-oriented decision analysis is a major aid in this transformation because it crosses the border between the graphic view of relationships that is very convenient for human beings and the explicit equations and numbers that are the province of present computers. To find a device that can readily be sketched by a layman and yet be so carefully defined that useful theorems concerning it can be proved by formal methods is rare. There is a great promise that the object-oriented decision analysis will be an important bridge between analysts and decision makers.

3.3 Object-oriented Problem Analysis and Structuring

Figure 3.1 shows the overall diagram and the general process for the methodology of object-oriented problem analysis and structuring. "Context" stands for the decision context in which the specific problem needs to be solved in a way of decision analysis. Context classes represent a category of decision contexts with similar features or in the same field of research such as a specific kind of natural resource management. The instances of the context classes are context objects, which indicate individual problem circumstances. People involved in decision analysis, such as decision makers, facilitators, domain experts, stakeholders, organisations, and other related parties are analysed. People analysis is mainly focused on the roles played by the people involved in the decision problem and decision making activities. Decision elements are entities that are included in the decision problem under consideration, those that take part in the process of decision making, and those that impact and/or are impacted by the decision made. Criteria, alternatives and decision makers are examples of decision elements. Decision element classes represent categories of decision elements with similar attributes. For example, an INCOME criterion class may be used to represent a composite set of various kinds of incomes in a decision problem. The instances of decision element classes are decision element objects, which indicate individual decision elements for a specific problem.

Problem analysis and structuring for a specific decision problem start from the identification of the decision context and the people involved. The decision context for the specific problem is a context object which is an instance of a corresponding decision context class for decision problems with similar features to the problem under consideration. The people involved in the decision problem are the people objects which are instances of the people classes generalised for the similar decision problems. Decision context classes and people classes are created out of the literature and past experiences. The context object and the people objects derived from their classes are used as a base of problem analysis and structuring. At the same time, other techniques of strategic analysis can also be utilised as supplementary approaches to complete and refine the context definition and the people analysis. New generic findings are put back to the context classes and the people class.

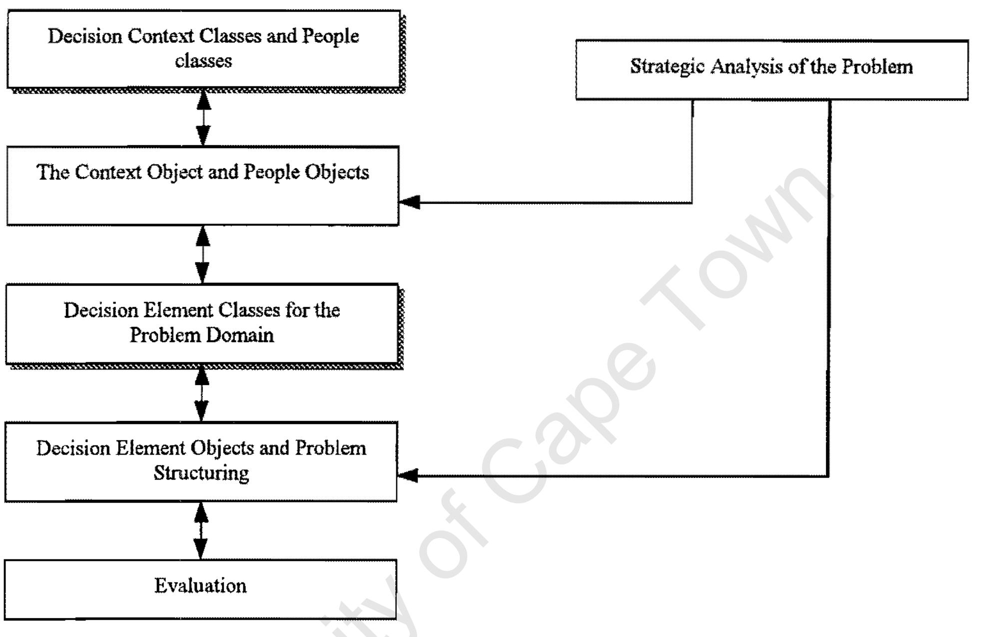


Figure 3.1: Object-oriented Problem Analysis and Structuring

The definition of the decision context and the analysis of the people involved can greatly contribute to the understanding of the problem and can lead to the further identification of decision elements. The decision context basically explains the causes of the results of the problem. It shows all the possible impacts of the decision problem and influences on the decision problem. It is a fundamental framework in which the decision problem exists. On the other side, people are critical and active players in the decision problem. They react to the changes of the problem environment according to their individual perceptions of the real world, which are based on their understanding of the problem. Decision elements, such as decision alternatives and decision objectives (criteria), are then defined to represent peoples' perception and expectation so as to make it possible to fulfil people's aspirations.

Decision elements are then identified. They are basically the definitions of the decision problem under consideration. Various aspects of the problem are represented as decision elements and analysed. Decision elements for the specific problem are objects (instances) of the decision element classes for the decision problem domain which contains the problems with similar feature to the problem under consideration. Decision element classes, just like other problem analysis classes, are obtained from the literature and past experiences. Techniques from other strategic analysis methods can be applied to refine the definitions of decision elements obtained from the decision element classes. New generic findings are put back to the decision element classes.

Problem structuring is then started based on the identification of various decision elements. Decision criteria, for example, are structured as hierarchies. Decision alternatives are generated out of the action elements defined. The communication mechanism is determined among various decision participants. Once the decision problem is structured, the evaluation and choice processes may start.

Strategic problem analysis can be carried out in many ways. Existing classes of various classes such as context classes and decision element classes can be a guide in carrying out the strategic analysis. Other approaches from outside object orientation can be used in a complementary or independent way to achieve the same objective. Soft methods from management science can facilitate the delimitation of the problem context, the definition and instantiation of classes. Cognitive mapping, for example, is used to illustrate the problem situation in the case study shown below. At the same time, hard methods may also be useful in coping with some specific aspects of the problem. Goal programming, for example, can be used to solve an optimisation problem based on well-defined values for a specific area that might be related to an attribute of the decision problem under consideration.

Context classes and people classes are the base for the knowledge-based analysis, and soft methods, such as cognitive mapping of behavioural representation for each person (or group), are a subsidiary means for the analysis. The analysis and structuring are carried out in an incremental way. That is to say, a later version of analysis and structuring is a refinement of a preceding step. In this sense, it is very similar to step-wise refinements in software developments. In fact, it is true that decision makers are constantly iterating amongst the stages of decision making, making revisions and bringing to their attention possible conflicts and inconsistencies in their preference as new insights are obtained and more knowledge about the problem is gathered in each interaction (French, 1984).

The methodology is human-oriented. People are analysed and modelled in the very initial stage in problem analysis and structuring. All the problem situations being addressed in decision analysis, whether in the public or in the private sector, whether in small firms or in giant organisations, all feature human beings in social roles trying to take "purposeful action" (Checkland and Scholes, 1990). People are immersed in complex action, which they are trying to make purposeful rather than instinctive or merely random.

The methodology utilises reuse, one of the major benefits from object-orientation, in decision analysis contexts. This is done through the instantiation of classes. Classes may represent the existing knowledge and experience of decision contexts and decision elements while objects are related to the specific problem situations. Reuse is however not the necessary basis for our method as the methodology can be used in the very first problem case, in which there are no existing classes. Strategic analysis may be helped by using existing domain knowledge and context classes such as environmental system classes, while problem structuring may be facilitated by using decision element classes, such as criteria classes and scenario classes. Without the existence of domain and element classes, the method can still be used effectively in both strategic analysis and problem structuring.

The situation of reuse in decision analysis problems is different from that in software engineering in that most decision problems are unique even though they might appear similar to the previous ones. Hence attention must be paid to the creation of reusable objects so that they will be able to allow the individual decision situations to utilise them directly. The next section discusses the object-oriented methodology of DSS development.

3.4 Object-Oriented DSS Development

Some aspects of DSS development can contain generic features although decision problems might be perceived to be specific and unique and most DSSs are designed to support a specific, relatively narrow decision problem. Future DSS development may maximise the re-use of previously created conceptual and technical components to deliver fast and reliable decision support.

Reaping the rewards of reuse requires both an understanding of the possible future use of a component and a commitment to build the component for reuse. Analysis and design reveal general and unique aspects of DSSs. The general aspects are eligible for reuse. These general aspects in the analysis of the domain and the development of DSSs are structured and categorised. Objects are used to store and retrieve the analysis and design.

Reuse therefore leads to domain analysis of decision problems and DSSs. Domain analysis for DSSs can help the understanding of the DSS world. Domain classes can be created as fundamental to the development of DSSs. They have significant potential for reuse not only within one organisation but also throughout the industry. This speeds up the analysis of new situations and the development of a new DSS.

A reuse repository is a central resource for the whole development process and can be searched for reusable components at each stage. An easily accessible repository is essential to support the reuse process through analysis, design, and implementation of DSSs. A two-level reuse repository is suggested by (Gossain, 1998). One level contains verified components that have undergone qualification and selection criteria to ensure that they are adequately tested and documented. The other level contains those components that have not been verified. These include not only reusable codes but also the storage medium for previous business models, process models, analysis models, design models, and other documentation from past projects. It is a place for shared knowledge and is a kind of classification of the system development knowledge. It, however, offers no guide in developing the system.

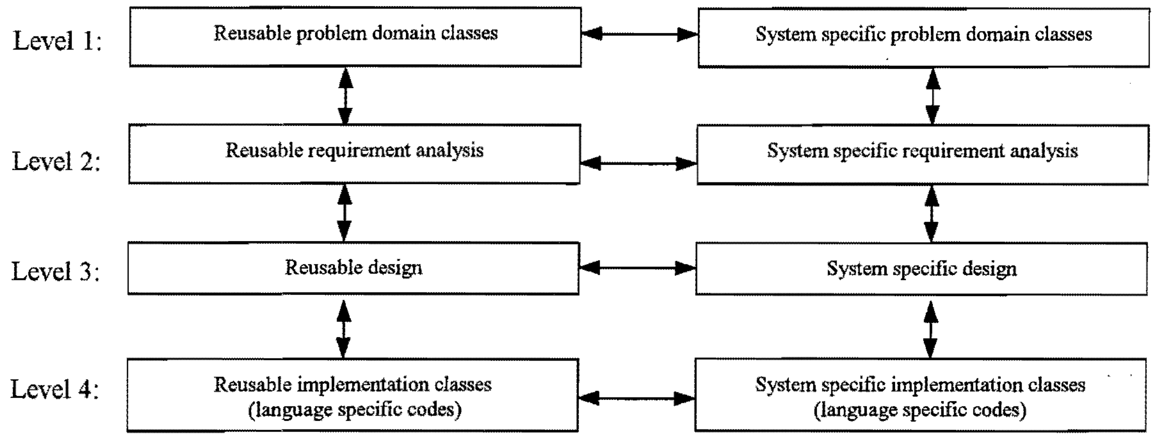


Figure 3.2: Reuse Repository Levels for DSSs

A four-level framework that consists of eight parts for the reuse repository is proposed as shown in Figure 3.2. It offers a guideline for the development procedure of DSSs. These levels include reusable problem domain classes, system specific problem domain classes, reusable requirement analysis, system specific requirement analysis, reusable design, system specific design, reusable implementation classes (language specific codes), system specific implementation classes (language specific codes).

Figure 3.2 actually shows a table with two columns and four rows. The four rows correspond to the four levels of reuse repository. The two columns respectively reflect the generic and individual system development procedures, which consist of problem domain analysis, requirement analysis, design and implementation.

The first column demonstrates the generic system models for DSS development. The system models may include those for reusable domain classes, reusable system requirement analysis, reusable design and reusable language specific implementation classes. The domain class model represent the basic knowledge of the decision problem domain and it constitutes the fundamental building material for DSSs. For instance, decision elements for the analysis and structuring of the decision problem are a type of domain analysis class. The system requirement model captures the general functionality for DSSs. It is about what DSSs do. The design model normally gives the information about the DSS structure and the system architecture. Reusable language specific implement classes are usually kept in a library called class library to store and retrieve system implementation classes, which are derived from the preceding system models and are the actual system components written in a specific programming language.

The second column shows the system development of a specific DSS, from problem domain analysis, requirement definition, and design to implementation. System specific building materials are derived from the models shown in the first column, by instantiating the generic DSS definitions according to a specific context. Although the ideal approach is to develop a system by using all these models, the system development can be speeded up by using any existing system model. The identification of the system requirements and domain classes, and the definition of the system design can be easily obtained under the guidance of existing models. The system can be quickly implemented in the programming language selected with the help of existing class libraries.

The first row indicates the relationship between reusable domain classes and system specific domain classes. Reusable domain classes can be obtained from the literature and experiences, and can be used when developing a specific system. Developing a specific system can in return contribute to the growth of the domain class model. The relationship between reusable domain classes and system specific domain classes is about the usage of knowledge and knowledge acquisition. The nature of this relationship also applies to those represented in the remaining rows.

The second row illustrates the relationship between reusable requirements and system specific requirements. Generic DSS requirements can be obtained from the literature and experiences. System specific requirements are then obtained by instantiating the existing DSS requirement model. On the other hand, the DSS requirement model can be created or adjusted by putting together the generic aspects of DSS requirement while doing individual system requirement analysis.

The third row reveals the relationship between reusable design and system specific design. Reusable design can be resulted from the analysis of the literature and experiences. Individual system design can be obtained from the instantiation of the

DSS design model and can in return be regarded as an experience in building the DSS design model.

The fourth row denotes the relationship between reusable implementation classes and system specific implementation classes. Reusable implementation classes are created as commercial or proprietary class libraries. These classes can then be borrowed or used as parent classes in generating system specific implementation classes. System specific classes can in return be generalised to form highly abstracted classes, which are stored in a class library, for the future usage purpose.

3.5 The Integration of Object-Oriented Decision Analysis and DSS Development

The two frameworks of the previous sections can be integrated as an object-oriented methodology for both decision analysis and DSS development. There are many common entities between them since DSSs are intended to support decision analysis, including problem analysis and structuring. The main classes created for problem analysis and structuring can be directly used in the development of DSSs. The problem context identified for the purpose of problem analysis and structuring still defines the environment for DSSs since DSSs are just computer-based system supporting some or all of the decision making processes for solving the decision problem. DSSs are part of the problem context. People classes resulted from the problem analysis and structuring are also the main players in DSSs since DSSs are designed for them to take part in and support the decision making processes. The decision elements identified for problem analysis and structuring are the basic type of knowledge and are part of the building material for DSSs. They are domain analysis classes.

Figure 3.3 shows the framework of the integrated object-oriented decision analysis and DSS development. Modelling of decision making can identify the most essential and fundamental classes and class relationships for the DSS analysis, and it constitutes the foundation for the domain analysis of DSSs. This is because a DSS contains only some of the entities involved in a decision making procedure. In other words, a DSS is only part of a macro system for decision support. An ideal DSS should include all the aspects of decision making to give it full support. Through the modeling of such an ideal DSS, a comprehensive set of primary entities, i.e., classes, can be obtained for both decision analysis and DSS development. These classes may include those of decision contexts, people, decision elements, etc., and may be used to carry out decision analysis in an object-oriented way as discussed in Section 3.3. Decision problems are defined with individual decision entities created from existing corresponding classes. A decision problem can be analyzed based on the defined classes possibly with the assistance of strategic analysis. On the other hand, these classes also help build a DSS model for the development of specific DSSs, which can in turn facilitate decision analysis by using computer techniques.

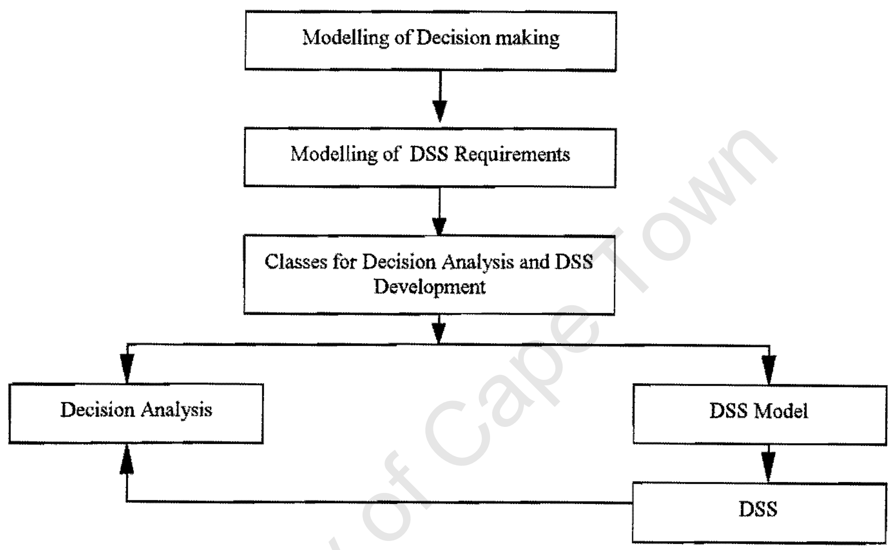


Figure 3.3: Object-oriented Decision Analysis and DSS Development

3.6 Conclusions

This chapter has discussed the methodology and the philosophy of object-oriented decision analysis and DSS developments. They are useful for many reasons. First of all, object-oriented decision analysis (especially in problem analysis and structuring) and DSS modelling offer an easy way to understand the decision problem and DSS development. This is because they model the real world naturally in a way that is focused on the conceptual side instead of the functional side of the world. Classes (objects) are the media for the representation of the concepts in the world. They are simple and can be transparent to all the world viewers. The interaction and clusters of these objects offer different viewpoints for observation of the world.

Secondly, object-oriented decision analysis and DSS development are able to improve the efficiency of decision analysis, especially problem analysis and structuring, and

DSS development. This is because they allow the easy usage of existing knowledge and the past experiences by utilising the classes of the generic aspects of problem analysis and structuring and DSS development. The problem context, decision elements, people involved, system requirements, system design and even language specific codes can be reused for future decision problems and DSSs with similar features.

Moreover, object-oriented decision analysis and DSS development provide a relatively comprehensive methodology for decision analysis and DSS development and can produce a better understanding of problem analysis and DSS development activities. They put an emphasis on people analysis beside their coincidental inclusion of the four main streams of problem structuring thought, i.e. the checklist stream, the definition stream, the science research stream and the people stream, as discussed in Section 2.4.4. At the very initial stage of analysis, people involved are identified and analysed. They are represented as objects with messages transferred to each other. Various decision participants or stakeholders from different geographical areas can then be modelled in the problem and the DSS without affecting the decision making activities.

In addition, flexibility of analysis is provided by the object-oriented decision analysis, especially problem analysis and structuring. Other methods of either soft or hard problem analysis can easily contribute to the problem analysis and structuring while object orientation is kept as the analysis basis. Decision problems can be analysed and structured starting from either generalised reusable generic aspects or individually specific entities of the decision problem. Decomposition of the problem by a top-down approach and composition of the problem components can be easily carried out since all the problem entities and the problem itself are regarded equally as objects in object orientation.

There is a need to point out that this chapter only introduced the main points of the philosophy and the methodology of the object-oriented decision analysis and DSS development. The frameworks, shown in the diagrams of object-oriented problem analysis (structuring) and DSS development, by no means prescribed strict procedures, entities and boundaries for problem analysis (structuring) and DSS development. They mainly served as a guideline for the applications of object orientation in decision analysis, especially in problem analysis and structuring, and DSS development.

The subsequent chapters will follow the methodology and philosophy of object-oriented decision analysis and DSS development by showing the detailed object-oriented modelling of problem analysis, the decision making procedure, and DSS development. The next chapter deals with the modelling of decision making in MCDM natural resource management decision problems, which offers an object-oriented method for problem analysis and lays the foundation for the analysis of DSSs.

Chapter 4

Modelling of Decision making in MCDM Natural Resource Management Decision Problems

4.1 Introduction

One of the objectives of this chapter is further to describe the philosophy and the methodology of object-oriented problem analysis (structuring) and DSS development (see Chapter 3). This chapter models the decision making context and decision making activities in MCDM natural resource management decision problems. This is actually an extension and a detailed application of object-oriented problem analysis (structuring) and DSS development in MCDM natural resource management decision problems. It offers an object-oriented method for problem analysis (structuring) for MCDM natural resource management decision problems, which also constitutes the basis for the analysis of DSSs for these decision problems.

The object-oriented method for problem analysis (structuring) for MCDM natural resource management decision problems is designed to fulfil four main requirements. First, the method should model the decision problem in a simple and transparent way and provide multiple viewpoints to the real world so as to make the problem easy to be understood. Secondly, the method should allow the accumulation and reuse of knowledge and experiences in the main aspects of decision making, including the problem domain, problem analysis and structuring, and other decision making activities. This can efficiently carry out the problem analysis and structuring for a specific natural resource management decision problem. Thirdly, the method should support group decision making in the natural resource management. Multiple interest parties dispersed geographically should be modelled in the decision making processes. Finally, decision elements of the decision problem should be represented as they are the basis of the definition of the problem, a kind of system building material of DSSs, and a basic form of knowledge that can be used in DSSs.

Besides its fulfilment of these four main requirements, the object-oriented problem analysis (structuring) method also meets other requirements for the object-oriented philosophy and methodology in decision analysis. For example, it allows the applications of soft and hard problem analysis (structuring) in the strategic analysis, which may be based on the existing reusable classes. However, not all the features are demonstrated here. Rigorous development and details of the utilisation of the approach to other phases of decision making and DSS development are left to further chapters.

Modelling of the problem solution procedure and the MCDM decision making procedure for natural resource management is another objective of this chapter. The modelling is designed to bring better understanding about the problem solution context and MCDM decision making activities. It is used to find out the generic aspects of decision making in natural resource management. These generic aspects are represented as reusable classes, which are regarded as an essential part of a DSS that is going to support some or all of the decision making processes. The attributes and behaviour of these classes are also preliminarily analysed in this chapter before their detailed analysis as decision analysis and DSS building material in the subsequent chapters.

This chapter is organised as follows. Decision problems for natural resource management are modelled in Section 4.2. Section 4.3 presents a general framework for decision problem solution and DSS context. Section 4.4 describes the identification of actors participating in decision problems, while the decision context and its elements are discussed in Section 4.5. The problem solution procedure for natural resource management is modelled in Section 4.6, while the modelling of the MCDM decision making procedure for natural resource management is discussed in Section 4.7. Section 4.8 proposes an object-oriented representation of the MCDM decision making procedure for natural resource management. Conclusions are contained in Section 4.9.

4.2 Modelling the Decision Problems for Natural Resource Management

A decision problem of natural resource management is about defining the situation, exploring the possibilities of future usage, and finding a solution. It does not automatically include the decision making procedure, which is discussed in the subsequent sections. The centre of the problem is the natural resource that is concerned by various stakeholders. The concerned natural resource has a usage cycle. The cycle starts with actors' perception, which is translated into action with varied results through formal or informal decision implementation. Some responsible actors learn from this experience to evolve the perceptions about the resource, refine its operations, and to co-operate with other actors. They continuously adapt to challenges and opportunities provided by every usage cycle. Figure 4.1 shows the usage cycle of the natural resource.

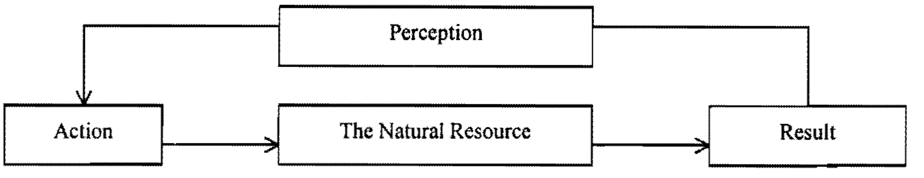


Figure 4.1: Natural Resource Usage

Systematic ways and scientific approaches can avoid the chaos in the usage of natural resources caused by uncooperative action by individual actors. The focus of natural decision problems is on the finding policy alternatives of consensus. It is about strategic planning based on various data, information, knowledge, understanding, and consensus. A proper understanding of the problem nature can ensure that various objectives from different interest groups are taken into play. An appropriate analysis approach is needed to ensure that various entities associated with the problem are sufficiently considered and analysed. In this way a fair agreement on the future action can be reached, and information about the future possibilities of some aspects can be obtained. In many cases of natural resource management around the world, however, proper approaches of analysis and decision making are not (or wrongly) adopted. Some decisions are made out of the rudimentary or instinctive analysis capacity of decision makers. But there is a growing awareness about the necessity and effectiveness of the application of decision analysis theory in these problem cases. Figure 4.2 shows the concepts of a natural resource problem under the analysis with some approaches.

Problem entities are the things, whether physical or not, that intrinsically exist in the problem. Problem entities are human, resources, and other things consumed and produced by natural resource usage. They contain the entities internal and external to the allocated natural resource. Stakeholders, finance, authorities, and the concerned natural resources are essential problem entities in natural resource management problems. The next section discusses the general framework for decision problem solution and DSS context.

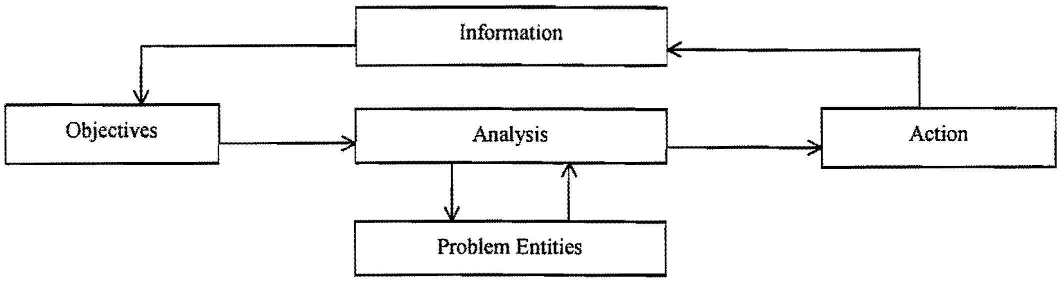


Figure 4.2: Natural Resource Decision Problems under Analysis

4.3 General Framework for Decision Problem Solution and DSS Context

A system of general perception, which is dedicated to solve natural resource management problems, manages processes and entities to achieve a purpose – to find a satisfactory policy alternative for all the parties involved. Such a system, which is called the macro-system, may include a broad range of entities of both real world and DSS objects, for example, as follows:

Real World Objects	**DSS Objects**
Decision workshops	Models
Interviews	Reports
Communication media	Decision making techniques (e.g., brainstorming and MCDM)
Human beings (e.g., facilitators, decision makers, domain experts, and other stakeholders)	Analytic tools

A DSS as usually perceived is a kind of software that comprises some entities in the system to facilitate the decision making procedure. The macro system is the complete MCDM process while the micro system is the DSS to be designed. In this chapter, we shall focus on interpreting the macro system in an object-oriented manner, in order to provide the framework for DSS design, which will be dealt with in Chapter 5. The analysis of the macro system leads to comprehensive understanding of the micro system, i.e. a DSS.

Figure 4.3 shows the integration of the macro and micro systems, and the context of a DSS for natural resource management decision problem. It can be seen as an extension of Figure 4.2 for natural resource decision problems under analysis since all the entities in Figure 4.2 are included in Figure 4.3. "Information" resulted from action in Figure 4.2 is represented simply by directed lines in Figure 4.3. "Objectives" and "analysis" are implicitly contained in "decision making procedure" and "choice" in Figure 4.3, while "problem entities" are contained in the "problem environment" and other entities in Figure 4.3. "Action" in Figure 4.2 are transformed as "implementation" in Figure 4.3.

As seen from Figure 4.3, the functionality of a DSS may contain some processes of the decision making procedure, choice making, and decision implementation. The DSS, either computer based or non- computerised, plays a critical role in the solution of the problem. Data and information from the natural resource for allocation and the problem environment and judgements from actors are input to this micro system to produce recommended choice and other insights as feedback.

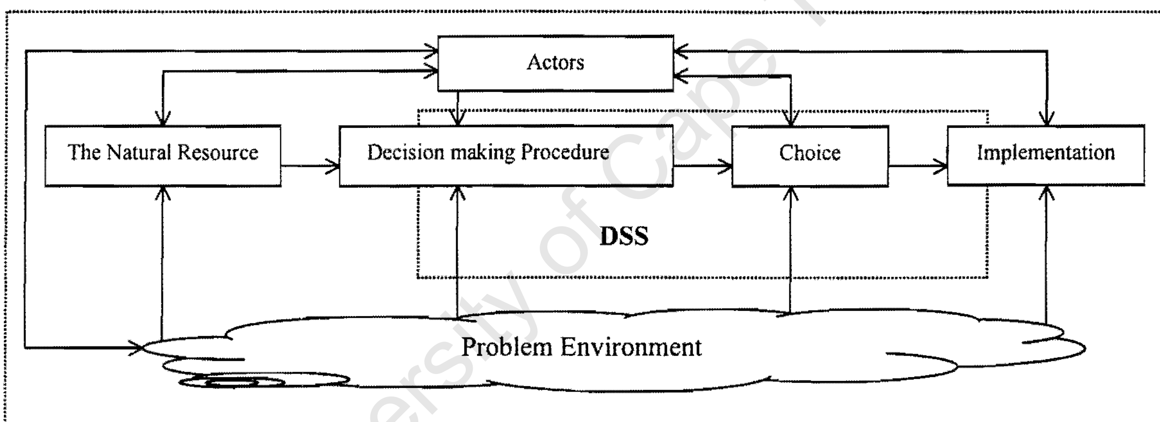


Figure 4.3: The Macro System for Solving Decision Problems

The figure also shows the framework of concepts, which includes the important and interacting theoretical perspectives that imply the need for a systematic thinking which, like SSM (Soft Systems Methodology) (Checkland, 1989), is composed of a flux of interacting events and ideas. The macro system is concerned with achieving objectives that lead to learning about a complex natural resource management problem and to purposeful action in the situation aimed at improvement. It is a learning system like SSM. There is no end point in the process of inquiry and action in the system unless actors choose to make one. New findings are made as the inquiry continues and better strategies can be found. The seven perspectives shown in the figure are about the actors, the natural resource, the problem environment, the decision making procedure, the choice, the implementation and the feedback.

The macro-system is regarded as a framework for the micro-system and is discussed in this chapter. The entities in both macro and micro systems can be interpreted as "objects" in an object-oriented philosophy. The object-oriented modelling of the micro-system, i.e., the DSS, is discussed in Chapter 5. The features of the natural resource are discussed in Chapter 1. Some perspectives of the macro-system are discussed in more detail below. The others are to be covered in the subsequent sections.

Actors are physical objects that have influence on the system under consideration when interacting with the system, for example, a physical person or group of persons who hold the same stake. The literature (Lootsma, Meisner and Schellemans, 1986; Stewart, 1988; Stewart and Brent, 1988; Mendoza, 1988; Glover and Martinson, 1987; Hallefjord and Jornsten, 1986; Sandiford,1986; Checkland, 1989; Checkland, and Scholes , 1990; French, 1995; Stewart, Joubert, Scott and Low, 1996; etc) has revealed some important classes of actors. A very important sort of person is the stakeholder. Stakeholders are those people, or groups of people, who either affect or are affected by the business in some way. Stakeholders of the system are actors, such as management, employees, shareholders, subcontractors, operators, users, consumers, and neighbours. Other actors may include domain experts, problem analysts, etc. The concept of actors here includes those of 'customer', 'actors', and 'owners' in SSM. We do not adapt the original concepts since in some situations not all of these three concepts exist. For instance, the concept of 'owners' may be difficult to define in the context of water resource planning. The "owners" may be the people, their political representatives, state structures such as governmental departments, or none of them (Stewart, Joubert, Scott and Low, 1996).

Actors play a central role in the system. Each actor in a natural resource management problem is held to have his or her own personal subjective view of the nature of the problem itself, the problem environment, the decision making processes, and the way of decision implementation. This is called "subjectivism" in the SODA (Strategic Options Development and Analysis) approach (Eden, 1989). Actors are involved in the psychological construction of the decision problem and the problem environment. The wisdom and experience of actors are an important element in developing decisions with which each participants feel confident. Actors play different roles in the system. A role is the responsibility of the actor at a certain time. Different actors take part in various tasks for exploring and solving the problem, including processes of decision making, making choice, and implementing the decision made. Special expertise is needed for some tasks like the application of analytical tools in the comparison of different decision alternatives. Some other aspects may merely need nothing more than expression of personal perceptions of the reality. The same actor in the system may have different roles. For example, a stakeholder of a local community in a water resource allocation problem might be a participant in the decision making procedure as well as an implementation agent of the final decision being made. Feedback from the consequences of the choice made and the implementation carried out affects the perception of actors about the natural resource for allocation and the decision problem environment.

The decision problem environment is the context in which the decision problem resides. It includes a collection of entities excluding from the allocated natural resource and its related actors and activities, as shown in Figure 4.3. No decision problems in natural resource management can exist isolated and aloof from the rest of the world. Decision making needs not only to respond the business and technological issues of the problem itself but also to deal with constrains, risks and influences from outside and exploit the opportunities and challenges that result.

Comprehensive classification of the problem environment entities is hardly reported in the literature. In the Strategic Choice Approach (Friend and Hickling, 1987; Friend, 1989), uncertainties about the working environment and other decisions and values are regarded overwhelming critical and important for decision modelling and strategic planning. Uncertainties are certainly a main consideration, if not more important, in the decision analysis of natural resource management problems. Other environment elements are often mentioned in environmental management (Keeney, 1988; Keeney, Renn and von Winterfeldt, 1987; Hallefjord and Jornsten, 1986; Romero and Rehman, 1985; etc) and complex societal problem handling (DeTombe, 1994, 1996), including conservation, economy, tourism, finance, human resource, technology, legal and political regulations, etc. These elements nevertheless belong to different aspects.

A classification of the problem environment elements is made based on the literature and the consideration of the problem solution as a system. Such a system consumes some resources as input and produces output under some constrains and conditions. Therefore, the elements of the problem environment are classified as four interacting components: resources, uncertainties, rules, and influence factors. Note that these four components are not necessarily inter-independent, but each of them may have impact on the others. For example, rules from governments may have constraints on resource usage. Figure 4.4 shows the categories and their interaction. Generally these components provide the basic sources of decision alternatives and perceptions of actors towards their subjective view of the problem. Resources mainly contain physical materials such as equipment, land and water, but non-physical matters such as technology, finance, authorities, time and human are also included. Apart from resources, decision making also needs to consider the uncertain aspects of the problem environment, rules that regulate the use and distribution of the resources, and influence factors on decision making activities. These issues are further discussed in Section 4.5.

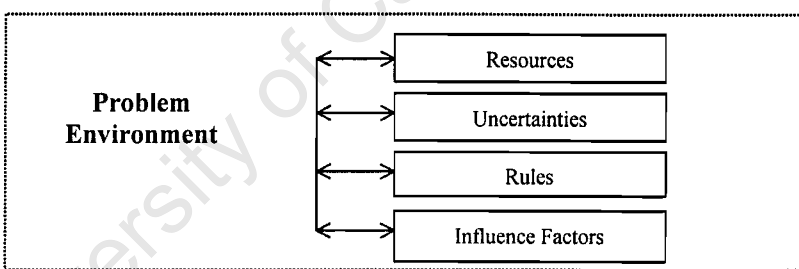


Figure 4.4: Problem Environment

To summarise, perceptions of actors about the problematical situations and ideas of actions to improve the situations are obtained via carrying out a sequence of continuing processes. A DSS can be a critical part to help actors analyse and solve the problem under consideration. The subsequent sections discuss the identification of stakeholders as well as general actors, influence factors, and the decision making context before the discussion of modelling general decision making procedure for natural resource management decision problem.

4.4 Actor Identification

Actors are usually discovered in the initial stage of decision analysis for natural resource management problems, in conversations with clients and domain experts, or by examining the previous analysis output of similar problems and existing systems. New actors will be found during the talks with clients. After interviews with these actors, more will be found. It is like SSM, in which new occupants of the roles 'problem solver' and 'problem owner' in the role analysis may emerge in the course of a study. In identifying actors, an incremental procedure has to be adopted in most problem situations as new actors turn up in the processes of analysis. There is not once-and-for-all analysis. Unlike some methods of problem analysis, we do not define actors into two divisions of internal actors and external actors in that for a system it is extremely difficult and sometimes becomes impossible to set a boundary, which depends on individual preferences and understandings. All the actors who have a stake in the system will have to be considered.

Actor classes in the decision problems in natural resource management represent categories of individual actors with same features. Based on the super actor classes, which include stakeholders, domain experts and problem analysts, discussed in the last section, and real practical cases (Lootsma, Meisner and Schellemans, 1986; Stewart, 1988; Stewart and Brent, 1988; Mendoza, 1988; Glover and Martinson, 1987; Hallefjord and Jornsten, 1986; Sandiford, 1986; Stewart, Joubert, Scott and Low, 1996; etc), actor classes of lower level of abstraction are identified for natural resource management decision problems. These actor classes are listed below:

- Domain experts, who offer domain knowledge in various aspects.
- Governmental organisations, who hold the policy and implement the final choice of the decision analysis, for example, the provincial water allocation committee.
- Non-governmental organisations or societies, who hold the same interest related to the decision to be made, for example, a mountaineering club.
- Local communities, who affect or are affected directly or indirectly by the decision, for example, the physical district where the natural resource is located, and a commercial company that uses the natural resource as raw material.
- Neighbour communities, who affect or be affected directly or indirectly by the decision and who are geographically apart from the site where the decision problem is, for example, a neighbouring district.
- Decision facilitators, who co-ordinate the whole process of the decision making.

These actors are further classified as domain problem solvers and stakeholders, which are shown in the actor-playing diagram below (Figure 4.5).

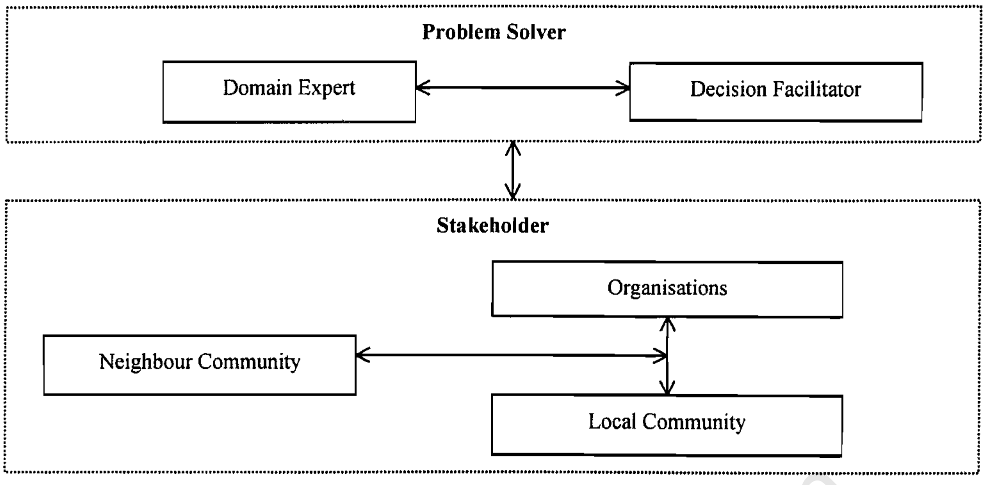


Figure 4.5: Actor-Playing Diagram for Natural Decision Problems

An actor-playing diagram shows the interaction of actors. Figure 4.5 shows the general actor-playing diagram for natural decision problems. It is a pictorial notation of an exploratory mode used to illustrate the inter-influences among different actors. Stakeholders provide necessary information and judgements to problem solvers. Problem solvers assist stakeholders to find possible policy alternatives, leading to a consensus of resultant action. In the processes of problem solving, decision facilitators need expert knowledge in regard to some specific areas while domain experts can obtain new insights from the results of decision making. Stakeholders may seek expert advice from domain experts. They interact with each other in different ways. Governmental organisations may have direct administration and control over the local communities and non-governmental organisations in terms of the usage of the concerned natural resources while they may have only indirect influence on the neighbour communities, which can be out of the administrative legitimacy of a government. In solving natural resource management problems, the "decision makers" are actually the governmental organisations or their entrusted agents, who may make up the final decision about the resource allocation. The diagram can be used as a way of communication among participants and analysts through the whole process of analysis.

The actor classes and the actor-playing stereotype are able to help the analysts identify actors and concentrate on the key actors in the analysis of a specific decision problem. These actor classes are instantiated to obtain individual actors. Domain expert actors are usually first identified in the problem analysis. After consultation with the experts, other actors are instantiated under the guidance of their respective classes. A virtual actor object called "other" is always defined as an actor for the macro system so as to keep the possibilities open. After the identification of actors, the actor-playing diagram for the decision problem under consideration can be generated by instantiating the stereotype diagram shown above. The interactions and relationships between these actors are therefore understood and discovered. Usually those actors who interact much more then the rest are identified as key actors. The analysis is then concentrated on these key actors. This can lead to efficient gain of insights and understandings about the decision problem. The following section discusses the delimitation of problem environments and the decision context in which actors exist.

4.5 Decision Context and Decision Elements

A decision context sets the boundaries of a decision problem along with its interfaces to its external world. A problem environment defined before is considered to be contained in a decision context. The delimitation of the problem environment for a specific problem involves the identification of influence factors, uncertainties, rules and resources (see Section 4.3 for detailed discussion). Actually these two terms, problem environment and decision context, may be identical literally. They are distinguished in order to emphasise the influences that problem environments can have on decision problems. The decision context is used here as a wide term to represent the problem sphere that contains the problem concerned, the elements included in the problem environment and their interfaces to the problem.

The classification of the elements and their sub-elements of the decision context serves as a guideline for analysts, to ensure that all aspects and issues related to the problem receive the necessary attention in every case. The classes of these elements and sub-elements can be used as templates for the problem analysis of a specific decision problem. The elements of the problem environment include influence factors, uncertainties, rules and resources. They are the main part of the decision context. In problem analysis, each of these environment elements needs to be defined before the interactions between them are analysed.

4.5.1 Influence Factors

Influence factors are those elements that influence the decision problem under consideration and the processes of decision making. Influence factors have impact on the final choice of decision alternatives, and the decision made will in turn have influence on them.

There exists a considerable body of literature regarding the influence elements of decision problems and decision making activities. Chankong and Haimes (1985), for example, list some system attributes, such as increase of agriculture production, enhancement of water quality, protection of wildlife, etc, for the consideration of a decision made in a water resource management case. Stewart, Joubert, Scott and Low (1996), for another example, suggest some other factors to be considered for a forestry land use decision problem, including employment, housing, personal well-being, tourism, industry, conservation, etc. These factors identified in the literature are then grouped into some broad categories, such as equity and economic growth and sustainability, which have been suggested for project or policy level planning (Faucheux and Froger, 1995; van Pelt, 1993). The coverage of these categories identified in the literature is however inadequate and it needs to include other aspects such as quality of life (Perrings, 1994). These aspects are merged into the overall categories. To summarise, influence factors in natural resource management problems are roughly classified into four categories: social, economic, political, and environmental factors (see Figure 4.6). As shown in Figure 4.6, the influence factors together with the decision problem are still within the boundaries of the macro system for decision problem solution. Some influence factors will in general conflict with others. For each of the influence factors, there are still more subclasses of factors to be considered. For instance, the environmental factor in forestry problems may be further divided into subclasses of the natural ecosystem and water run-off patterns. These factors may be built into decision criteria for systematic analysis at a later stage.

Like actors, the influence factors are also usually discovered in the initial stage of decision analysis, in conversations with clients and domain experts, but mainly by examining the previous analysis output of similar problems and existing systems. Primary influence factors may be identified after a brief consultation with the domain experts. More factors will be found during the talks with domain experts and your clients. An incremental procedure has to be adopted in most problem situations as new factors turn up in the processes of analysis.

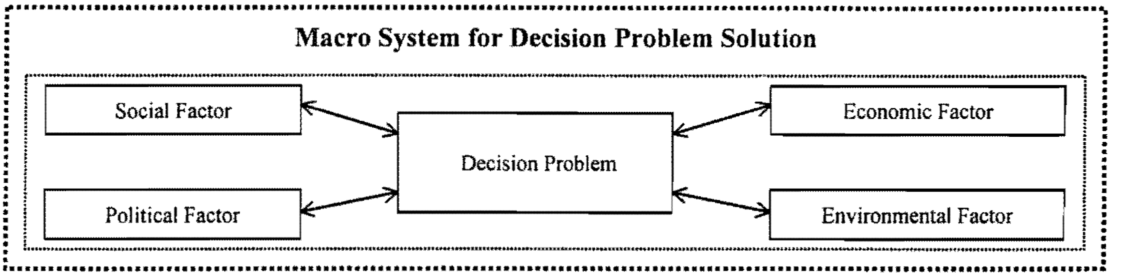


Figure 4.6: Influence Factors for Natural Decision Problems

4.5.2 Uncertainties

People will experience personal feelings of uncertainty at many moments in a decision making process, and that these feelings will change continually as the process proceeds. People may feel uncertain as to whether a specific national law regarding the natural resource usage could be changed. They may feel uncertain as to the physical boundaries of certain natural resources. They may feel uncertain as to whether or not particular decision areas should be seen as interconnected. They may feel uncertain as to whether particular options or combinations of options should be considered feasible. They may feel uncertain as to the terms in which particular comparison areas should be formulated. They may feel uncertain about their own judgements and preferences in the comparison of decision alternatives. French (1995) defines uncertainty to include the various forms of uncertainty that may arise from imprecision or ambiguity or lack of clarity.

At least three kinds of uncertainty can be observed in natural resource management problems. The first kind of uncertainty is the imprecision of human minds and their expression about subjective judgements such as the importance of water supply to downstream areas. There is no formal probability theory formed for solving this kind of uncertainty. The second kind of uncertainty is about the unreliability of data due to lack of sufficient information. Supply of additional information is the key to uncertainty solution. The third kind of uncertainty is from unknowable random processes, for example, the weather condition in the next 20 years. Statistical distribution theory may be applied to solve this kind of uncertainty.

Three categories of uncertainty are identified in Strategic Choice Approach (Friend and Hickling, 1987; Friend, 1989) as uncertainties about the working environment (UE for short), uncertainties about guiding values (UV for short), and uncertainties about related decisions (UR for short). In the UE direction, participants in a process of strategic choice may experience personal doubts, or may differ among themselves, as to the assumptions they should make about external circumstances or trends. In the UV direction, they may experience doubts or disagreements as to the values that should influence them, especially when they are seeking to compare alternatives across different comparison areas which reflect the concerns of diverse interest groups. In the UR direction, they may have difficulties agreeing what assumptions to make about the choices that are expected to be made in the future in other decision areas outside the current scope of the problem on which they are working. They might conceivably have some influence over these decision areas, even if that influence may be quite limited or indirect. Attempts to manage the current state of uncertainty involve the demands for more information, clearer objectives, and more coordination over the current decision situation.

A number of different categories of uncertainty, which need to be considered in the three major steps in conducting analyses, are suggested by French (1995) along with different analysis models, including descriptive, normative and prescriptive models. A descriptive model is a conjectured picture of reality, which mirrors possible relations between possible objects or classes of objects in the external world. A normative model suggests how we might think, choose or act. It seeks to capture a possible set of principles which we might wish our thinking, our judgements, our decision making and our behaviour to obey. A prescriptive model is the amalgamation of descriptive and normative models in analyses. It offers a means to bring understanding of our beliefs, perceptions and preferences in relation to issues before us in a particular inference or decision, while in descriptive analyses, the understanding is of the world about us; and in normative analyses it is of norms of behaviour. The three steps in conducting analyses are modelling, which constructs a set of models for the real world, exploration, which explores the models, and interpretation, which interprets the conclusions of the explorations into guidance for real-world beliefs, inferences and decisions.

Uncertainties expressed during modelling include uncertainty about might happen or what can be done, uncertainty about meaning/ambiguity, and uncertainty about related decisions. Uncertainty about what might happen or what can be done is suggested as one type of uncertainty which cannot be modelled with normative and descriptive models because it is impossible to identify all client's concerns and think of all possible strategies. As to uncertainty about meaning/ambiguity, a descriptive model, which is behind fuzzy mathematics and possibility theory etc, of the ambiguity and imprecision present in a third party's (some one distinct from the analyst and the clients) statement is useful. The need for normative models is questioned because a methodology with an emphasis on modelling, rather than resolving, ambiguity and imprecision, cannot serve this aim. Uncertainty about related decisions can be handled by including the decisions in a decision tree, influence diagram or other method of representing interrelated decisions. Their interactions can further be investigated via dynamic programming. This is under the condition that the analyst is supporting a coherent group of decision makers. If others are involved, the form of the models may be drawn from the literature of game theory.

Uncertainties expressed during exploration of the models consists of uncertainty arising from physical randomness or lack of knowledge, uncertainty about the evolution of future beliefs and preferences, uncertainty about judgements, e.g. of belief and preference, and uncertainty about the accuracy of calculations. For uncertainty arising from physical randomness or lack of knowledge, there seems to be general agreement that it should be modelled using mathematical probability. Uncertainty about the evolution of future beliefs and preferences is pointed out as an area that needs much more research and thought, and there is a need for normative models to help analyses of strategies which may lead to consequences in the distant future. For uncertainty about judgements, axiom systems which effectively lead to interval methods are regarded to be able to avoid precise judgmental input. Symbolic values are introduced along with bounds on the values. Sensitivity analysis, for example, does not require precise bounds on the ranges of judgmental quantities. Uncertainty about the accuracy of calculations is considered as a less concerned issue. O'Hagan (1992) and Mockus (1989) are suggested as some ways of modelling this uncertainty.

Uncertainties expressed during interpretation contains uncertainty about the appropriateness of a descriptive model, Uncertainty about the appropriateness of a normative model, and Uncertainty about the depth to which to conduct an analysis. For uncertainty about the appropriateness of a descriptive model, various schools of statistical inference have suggested ways (e.g. hypothesis testing, analysis of variance, and mixtures of models) of making a choice among a variety of models. However, this form of uncertainty cannot be avoided since there will always be difference between a model and reality. For uncertainty about the appropriateness of a normative model, the exploration of the implications of each normative model, to see where they differ, seems to be the only viable approach for modelling and analysis. For uncertainty about the depth to which to conduct an analysis, it is thought to be possible to keep refining the models used in an analysis, introducing mode and more subtleties. Phillips (1984) presents a useful way to explore the models by means of sensitivity analyses, by residual plots, and by using the methods proposed by Box (1980) until no further insights are gained and no new doubts arise, or no thing more can be done.

French (1995) shows different objectives of uncertainty modelling even though the list of categories of uncertainty may not be fully explored. It is also pointed out that in prescriptive analyses there are some forms of uncertainty which should not be modelled, and rather the analysis should seek to resolve or reduce the uncertainty through its modelling and exploration of other aspects of the problem.

It is important to view all areas of uncertainty during the whole decision making procedure even though some areas do not need to be modelled. Uncertain aspects of decision problems should be considered from the initial phase of analysis. A working list of uncertainty areas is built up progressively as feelings of uncertainty may surface at virtually any moment in a process of decision analysis. Participants are asked what the areas of uncertainty are. Uncertain areas can be identified by examining whether some suggested elements of a problem should be considered or expressed as decision areas at all. Some uncertain aspects identified may seem to lie largely outside the sphere of the macro system, in which case they might perhaps seem better eliminated as irrelevant areas. Some other uncertain aspects may be better expressed either as comparison areas or as decision alternatives, in which case individual approaches of uncertainty analysis are needed to deal with their management and resolution. Uncertainty areas from the initial analysis can always be transferred separately for further consideration when the decision making analysis proceeds. Possible outcomes of a certain uncertain area may be represented as uncertain scenarios so as to keep options open for later resolution. With the assistance of more information and additional action; the situations will become clearer than before at the later stage of analysis. Uncertainty will be resolved with satisfaction as a result. This is very similar to the exploratory approach used by the Strategic Choice Approach (Friend and Hickling, 1987; Friend, 1989). Both the scenario representation approach proposed here and the exploratory action approach offer a hope that current feelings of uncertainty can be significantly reduced within some uncertainty areas before decisions have to be made.

In the Strategic Choice Approach, the opportunities for taking immediate actions in some decision areas while deferring choice in others are captured as action schemes. Any exploratory action invariably takes some time to carry through, and can thus imply delays of to a more or less serious extent in the taking of those decisions that are designed to inform. The consideration of what to do about particular uncertainty areas can bring concerns about the timing of choices in different decision areas directly to the fore. After the analysis of a set of immediate decisions covering both actions and explorations to reduce uncertainty and then a set of proposed arrangements of deferred choices and contingency planning with a future decision space, commitment packages are designed to indicate the incremental steps towards commitment through time, reflecting resource constraints, urgencies and priorities of the specific decision setting. In the scenario representation approach, on the other hand, only exploration into the options of uncertainty outcomes is carried out. Options are closely examined and the consequences of following these exploratory options are weighed up. No actual action will be taken in the decision areas before the decisions are finally made. This has been proved necessary in the natural resource management decision analysis due to the features of this kind of decision problem, especially because of the irreplaceable and, in most occasions, non-undo and irredeemable characteristics.

4.5.3 Rules and Resources

Rules and resources in the problem environment are relatively easier to identify in comparison with the identification of influence factors and uncertainty areas. Rules are the regulatory governmental acts and contracts between agencies that may control and influence the use and distribution of various resources. They can be identified by examining the relevant laws, contracts and other legitimate documents. Resources in the problem environment are mainly referred to physical materials such as equipment, land and water, and non-physical matters such as technology, finance, authorities, time and human, which are consumed or produced by the processes of decision making and the consumption of the concerned natural resources. They are identified after considering what are needed for various processes of decision making and for the usage of the concerned resources, and what are the products of such usage.

4.5.4 Decision Context Diagram and Decision Elements

The decision context diagram, as shown in Figure 4.7, is a graphical representation of the decision context, which set the context and the boundaries of a problem. The context diagram for a specific problem case can be derived from this general context diagram. Relationships between different entities are included and showed by arrows.

Actors, influence factors and other elements can be "external" and "internal", but they are not distinguished as "external" and "internal" at the initial analysis. They may be further analysed and considered as decision elements in the decision making. All these elements can be shown as objects, while their interfaces are shown as messages to and from objects.

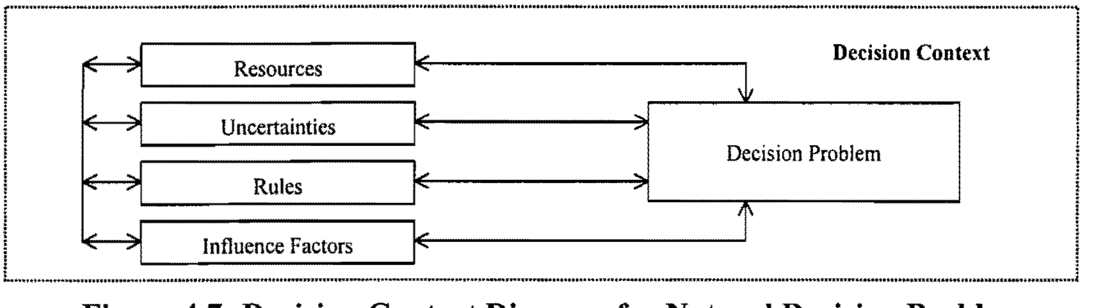


Figure 4.7: Decision Context Diagram for Natural Decision Problems

Decision elements are the entities, whether physical or not, that intrinsically exist in a decision problem and are related to the decision making processes. They are the basis for decision analysis, and the collection of their definitions defines the decision problem under consideration. Some problem entities discussed in Section 4.2 or their variations, such as stakeholders and constraints of resources, and many entities implicated, such as choice, implementation and uncertainties, in the macro system for decision problem solutions, are decision elements. Decision elements also include the entities that take part in the process of decision making and those that impact or be impacted by the decision made. Criteria, alternatives, and decision makers are examples of these kinds of decision elements. Decision element classes represent abstracted categories of decision elements with same features. Decision element objects indicate individual decision elements for a specific problem.

Decision elements are abstracted as classes, which represent a category of decision element objects with same features. Each aspect of these abstractions is a potential part of the vocabulary of problem analysis (structuring) and the DSS that will implement part or all of the processes of decision making. Decision elements are mainly obtained from the MCDM terms in the literature. Many terms for MCDM have been defined in the literature (von Winterfeldt, 1980; Goicoechea, Hansen and Duckstein, 1982; Chankong, Haimes, Thadathil and Zionts, 1985; Steuer,1986; Zionts and Lotfi, 1989; Bana e Costa, 1990; Stewart, 1992, Stewart, Scott and Iloni,1993; Roy, 1999; Belton, 1999; etc). Stewart, Scott and Iloni (1993) define most of these common terms, such as decision (policy) scenario, attribute, criterion, utility, objective (goal) and solution. Belton (1999) lists a few elements related to a MCDM process, such as goals, constraints, stakeholders, values, key issues, alternatives, criteria, etc. Although there is not unanimity in the literature on these terms, the most fundamental are identified and classified.

The essential classes for decision analysis of a natural decision problem are therefore defined as including those of decision attributes, criteria, alternatives, constraints, uncertainties, facilitators, domain experts, stakeholders, decision makers, and other decision participants. These decision element classes represent the things that are important to shareholders in the problem and the participants in the procedure of decision making. They are discussed in detail in Chapter 5 and 6 together with the modelling of a comprehensive set of decision elements.

Based on the identification of the essential decision elements, the problem context and actors, the problem solution procedure and the decision making procedure using MCDM are modelled. This is covered in the subsequent sections.

4.6 Modelling the Problem Solution Procedure for Natural Resource Management

As discussed in Section 4.3, systematic ways and scientific approaches are needed in finding a solution for natural resource management problems to avoid severe consequences caused by uncooperative action by individual actors. Sufficient consideration of various aspects of the problem is also needed to reach a consensus on future action among different interest groups. These interest groups must communicate with each other and work together to find a solution for the problem or an agreement on solving the problem. Figure 4.8 shows a diagram for the problem solution procedure, in which various communities and governmental organisations co-ordinate by themselves with the facilitation of decision analysts.

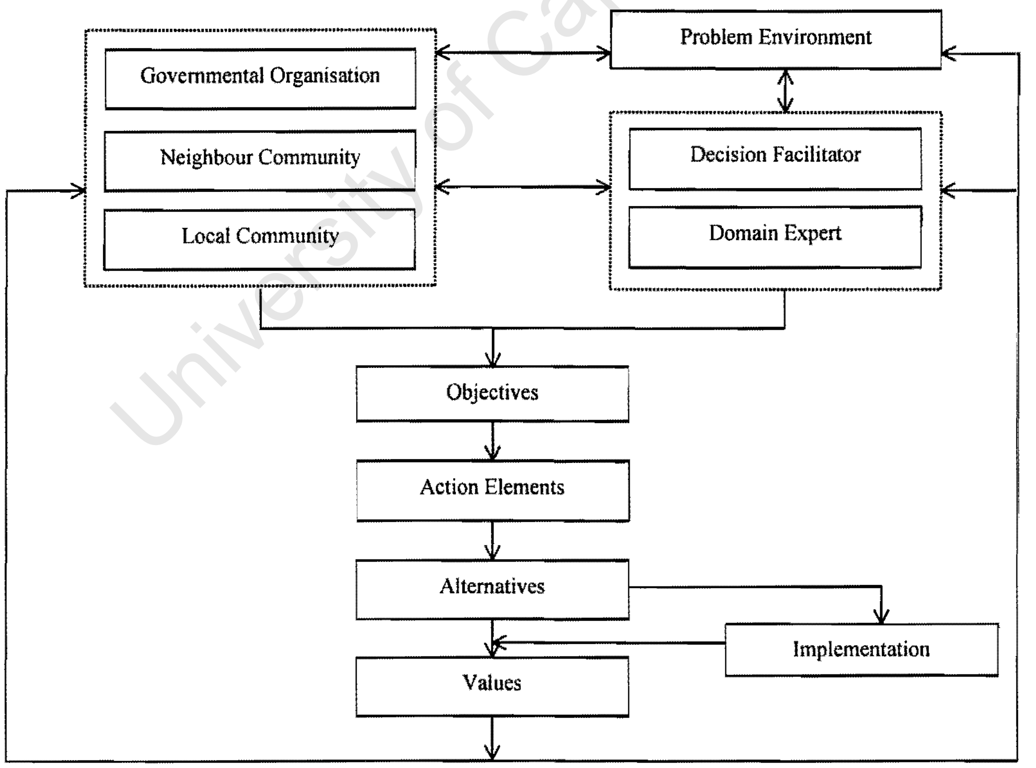


Figure 4.8: Problem Solution Procedure for Natural resource management

The figure shows the community involvement in the problem analysis, and it constitutes a basis for modelling the decision making procedure using MCDM.

Individual groups may consider each other's interests and are making some compromise while they may have no idea about what approaches of decision analysis they are using. Three perspectives of the macro system, the decision making procedure, the choice, and the implementation, as well as the actor-playing diagram identified in Section 4.4 are actually included in this solution procedure. The problem may be analysed and solved during the processes of identifying various decision elements, such as objectives, action elements, and alternatives, and observing the values caused by the choice and the implementation of the decision. Actors, especially stakeholders, may first of all have their own objectives in their minds for their own business. They may think then what kinds of action, which are called action elements, they can take to achieve their specified objectives while at the same time the identification of action elements constructs the decision alternatives, out of which actors may choose one that all participants are satisfied. During this process of decision element identification, some insights may be obtained, and specially some further implementation of the decision made can lead to useful values that can feed back to various organisations, communities, and individuals. The discussion of the decision making procedure using the theory of MCDM is followed in the next section.

4.7 Modelling the MCDM Decision Making Procedure for Natural Resource Management

MCDM has been extensively applied in the management of various kinds of natural resources. The literature of MCDM (Romero and Rehman, 1987; Stewart, 1992; Mendoza, Campbell and Rolfe, 1986, 1987; etc) has revealed that natural resource management problems constituted a substantial proportion of applications considered. Many of these environmental projects have succeeded, and a common understanding is being reached among the academic personnel that environmental problem are multi-criteria, and can be managed with MCDM if properly applied. Real practical cases(Lootsma, Meisner and Schellemans, 1986; Stewart, 1988; Stewart and Brent, 1988; Mendoza, 1988; Glover and Martinson, 1987; Hallefjord and Jornsten, 1986; Sandiford, 1986, etc), however, vary in terms of benefits that MCDM has brought about mainly due to how the MCDM techniques are used in practice. A general framework of MCDM decision making concepts can help the understanding of the basic considerations of MCDM application in natural resource management.

4.7.1 Conceptual Framework

A conceptual framework, which is shown in Figure 4.9, is proposed for the general MCDM decision making procedure for natural resource management. Figure 4.9 shows the group structure of decision participants and the group interaction. According to Phillips (1988) a group decision making procedure should be able to help structure the thinking of the group, and deal with group dynamics. Its method of operation should be understandable to participants and it should be flexible. In the context of resource allocation problems with high levels of conflict and antagonism, these features seem to be especially important. Different viewpoints from various actors, including stakeholders, a decision facilitator, domain experts, which are called decision participants, can be observed as well as an overall perspective of decision making processes. Group decision participants communicate with each other under the guidance of a facilitator (a decision analyst), who is skilled in the processes of group discussion and decision analysis, and has some expertise (at least basic knowledge) in the context of the problem at hand. The facilitator assists the participants in brainstorming the problem and structuring the decisions facing the group. The problem structure and value judgements come from the group.

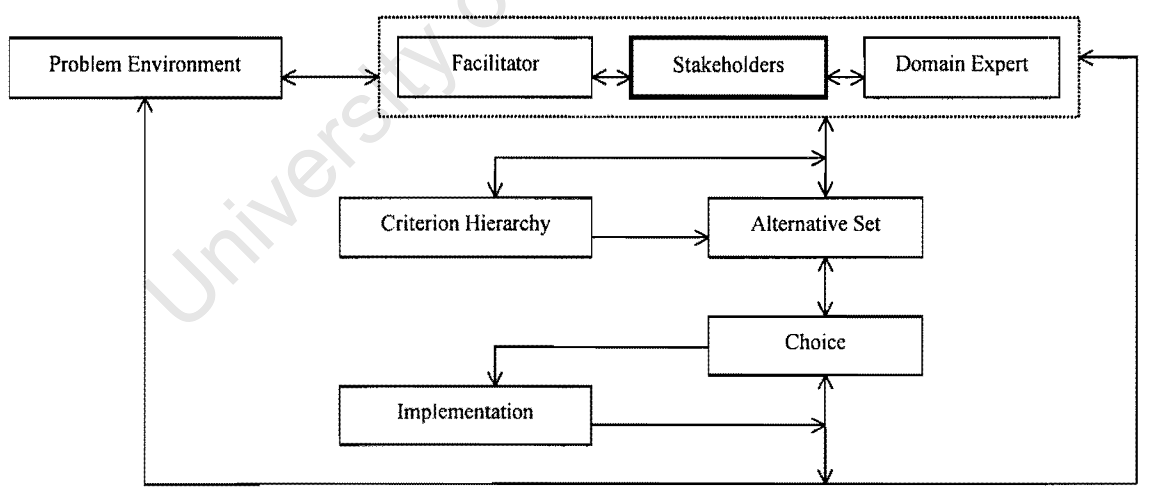


Figure 4.9: General MCDM Decision making Procedure for Natural Decision Problems

All participants, however, may carry out almost the same kinds of activities in the decision making procedure although the main players are stakeholders, which are represented with a bold rectangle. The facilitator, stakeholders, and domain experts may for their own purpose identify decision alternatives and criteria, construct an alternative set and a criterion hierarchy, and make a choice. For example, a domain expert may need to make a decision in regard to a specific knowledge before he or she can provide reliable information to the facilitator and stakeholders. The final choice about the allocation of the concerned natural resource, whereas, is ultimately determined by the stakeholders, who express their own preferences and seek a consensus among each other to reach an agreement on the choice.

Criteria are usually constructed hierarchically. Keeney and Raiffa (1976), and Von Winterfeldt and Edwards (1986) show that it is possible even in complex problems to structure management criteria hierarchically. At the top of the hierarchy is a general statement of the overall criteria (e.g. economy), which need to be progressively better defined at lower levels of the hierarchy to clarify the vagueness of the meanings of the broad criteria. This refinement of criteria will continue until they become very specific (e.g., employment rate) at the lowest level. These lowest objectives need to be operationally meaningful in the sense that a decision alternative can be more or less unambiguously judged according to this attribute.

Creating a hierarchy of evaluation criteria is advantageous, because it enables the user to disaggregate highly complex and generic criteria into their measurable components. Expert judgement and existing data are likely to be more effectively incorporated in guiding evaluations of these more concrete criteria. In addition, the clustering of criteria within hierarchies simplifies across-criteria comparisons. The user systematically judges the relative value of each alternative on each criterion, and then judges the relative contribution of each criterion to the whole. Working through this systematic procedure permits the user to make a small number of relatively simple judgements to determine the relative value of the alternatives. The necessity for the user to make unaided the highly complex, and often unreliable, overall judgement of preference between the alternatives is thus avoided.

It is noted that objectives in Figure 4.8 are replaced by criteria in Figure 4.9. Objectives represent what are expected to achieve in a decision problem, while criteria represent the corresponding points of view for the purpose of decision alternative comparison. Objectives are actually quantified representations of criteria, and are usually expressed in a form of measurable values. The identification of specific objectives and criteria is considered as a necessary pre-requisite to further MCDM analysis by Stewart, Scott and Iloni (1993).

Two important kinds of objectives, fundamental objectives and means objectives, are defined by Keeney (1992), and the construction of objective hierarchies is suggested as very useful in decision making. A fundamental objective indicates an essential reason for interest in the decision situation. A means objective implies that more fundamental objective can be achieved through this in the decision context. . Fundamental objectives correspond to criteria in Figure 4.9, and they are essential to guide all the effort in decision situations and in the evaluation of alternatives. However, means objectives can also be very useful for developing models to analyse decision problems and for creating alternatives through objectives networks and hierarchies.

Two types of objective constructs: means-ends objectives network and fundamental objectives hierarchy, are proposed by Keeney (1992). A means-ends objectives network is a graph with arrows to mean influences between different types of objectives. In such a network, the lower level objectives under any higher level objective can demonstrate how the higher-level objective can be better achieved. A fundamental objective hierarchy is constructed with fundamental objectives. It corresponds to the criterion hierarchy in Figure 4.9. In a fundamental objective hierarchy, the lower level objectives under any higher level objective represent the important aspects of the higher level objectives. For each fundamental objective, attempts are made to specify all the means to achieve that objective. This process is particularly useful to stimulate thought about the range of alternatives that may influence the achievement of the fundamental objectives. Obviously besides other advantages, these objective constructs help identify missing objectives, since logical concepts of the specification process can fairly easily identify holes in a network or hierarchy. The list of objectives is then converted into a hierarchy of criteria, which is used in the evaluation of alternatives in the decision making.

The decision making procedure shown in Figure 4.9 is iterative. Decision participants may first identify various alternatives as well as objectives (criteria) as a result of brainstorming. A choice is made out of the evaluation of alternatives according to the criterion hierarchy constructed by using MCDM techniques, such as those from Multiattribute Utility Theory (MAUT) (Keeney and Raiffa, 1976; von Winterfeldt and Edwards, 1986), Analytic Hierarchy Process (AHP) (Saaty, 1980), the Outranking approaches (Roy, 1990), goal programming, etc. An implementation of the decision made may be (and may not be) planned, and action ensued may have impact on the problem environment and will be reflected on the perceptions of the participants on the problem. Feedback in fact can be gained at many moments of the decision making procedure, and there is always possible to reiterate a process that has been done. This situation is indicated in the figure with two-way arrowhead lines.

The conceptual framework provides fundamental specifications for a DSS, which is based on MCDM and can facilitate actors to make decisions for natural resource management decision problems, as it identifies the essential decision elements and the fundamental functions involved in the decision making processes. However, beside a general illustration of the MCDM decision making procedure, there is a need for decision making activities to be analysed in detail to obtain a comprehensive understanding of decision making processes.

4.7.2 Decision Making Activities

Decision making activities are shown in Figure 4.10, an activity diagram that models the sequential and concurrent steps in the flow of control from activity to activity in the decision making procedure. Activities are represented by dotted rectangles. Initial and final states are indicated as a solid dot and a small square respectively. Control flows are illustrated with directed lines. Bold sold lines indicate concurrent forks and joins. The activities in different rectangles may be independent and concurrent. The activity takers, i.e. stakeholders, the facilitator, and domain experts, are listed in the head of the figure.

The activity diagram illustrates the interactive activities of decision participants. Under the guidance of the facilitator, all decision participants will brainstorm the decision problem concurrently via various techniques such as decision conferencing and interviewing. One of the substantial results from the brainstorming is some basic decision elements, such as action elements and criteria, identified by all participants. Domain experts and the facilitator may check and validate these decision elements identified for their completeness, accuracy, correctness, consistency, etc. For example, they may need to check if a certain species needs to be protected which may be extinct due to the changed land conditions as a result of land allocation decision. If any problem arises, the decision making then may need to clarify it before moving to the next stage.

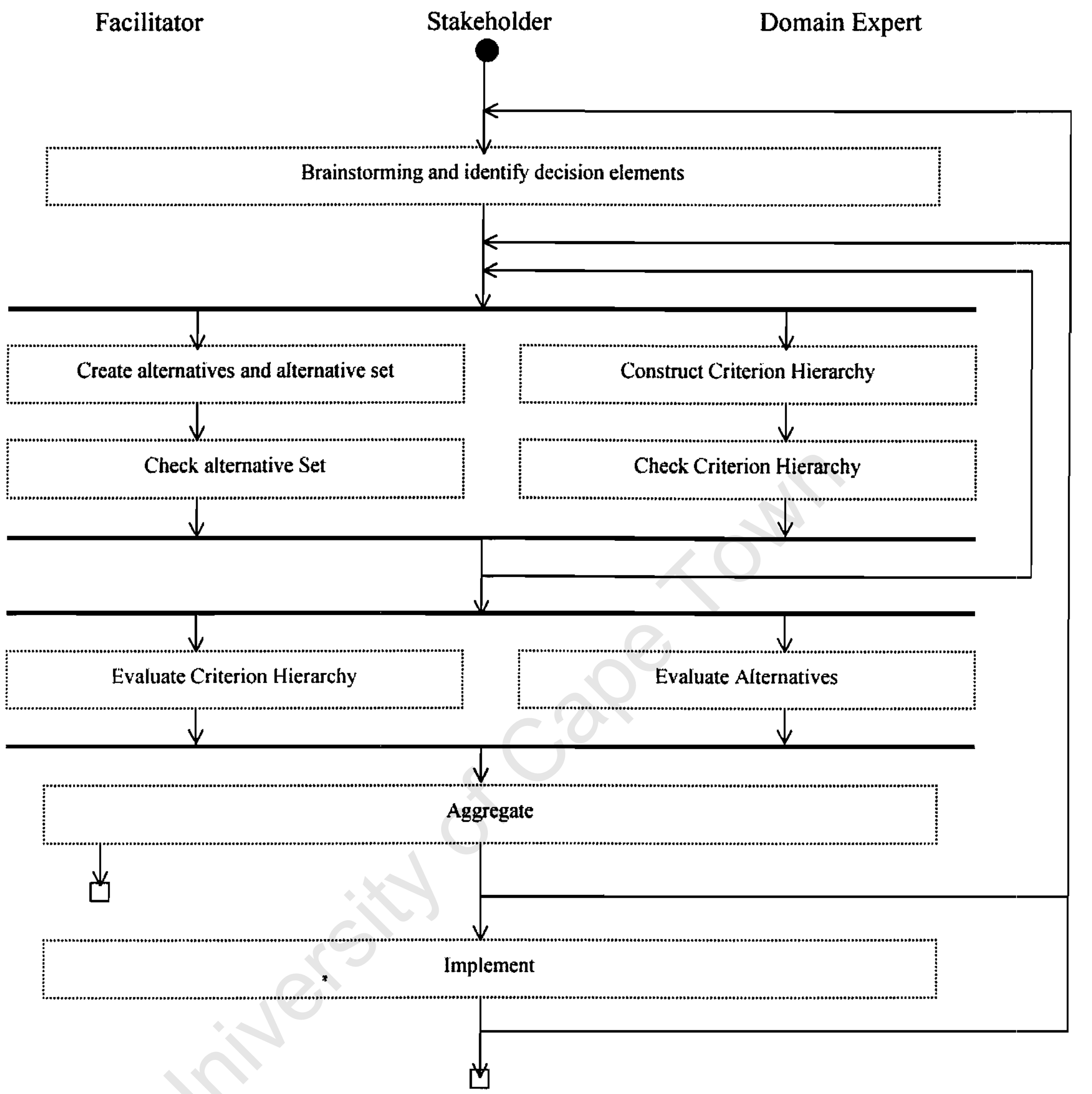


Figure 4.10: Activity Diagram for Decision Making Procedure

After the identification and validation of some basic decision elements, all participants may construct a criterion hierarchy for their own purpose, and may concurrently generate alternatives and alternative sets at the same time. Domain experts may need to carry out their own tasks of decision making to obtain knowledge about some issues that are critical to the overall decision making about the allocation of the natural resources. A collective alternative set for the overall system will be finally formed out of the alternatives generated by various stakeholders. The alternatives and the criterion hierarchies generated may be checked and validated concurrently. Criterion hierarchies need to be checked by domain experts to ensure that all the concerns are taken into consideration. Alternatives may be checked for some issues as well, such as their feasibility and inclusion of sufficient action elements for the problem under consideration. Again, if any problem arises, the decision making procedure then may go back to the previous stages, especially to the point where alternative and criterion hierarchies are generated, to redo some of the activities of decision analysis.

After the generation of the alternative set and the construction of criterion hierarchies, criteria are evaluated, and alternatives may also be concurrently compared by individual stakeholders and domain experts according to their criterion hierarchies. Results of various kinds, such as a chosen alternative and ranked alternatives, are then aggregated for stakeholders and for domain experts that need to make their own decision about some relevant issues before the overall aggregation is made, which integrate the results from all stakeholders to build a holistic evaluation. This may be the end of the decision making. However, insights about the problem and decision making activities may be obtained and the procedure may go back to any point of the previous stages to reiterate the whole processes. In addition, it is still possible to plan an implementation for the decision made with the potential participation of all decision participants. More insights may be gained from the planning and from the action taken thereafter. A new round of decision making will then take place. The decision making procedure is iterative as there are many possibilities to go back to the previous stages at various points.

The segments separated by the bold solid lines in the activity diagram actually represent iterative processes of the decision making procedure. Five processes can obviously be observed from the diagram. Brainstorming and identification of basic decision elements are carried out in the first process. This process is called problem analysis as it explores different aspects of the problem trying to obtain understandings. The second process deals with the generation of alternatives and the alternative set, and the construction of criterion hierarchies. This process is called problem structuring as it puts various elements together in structures for further decision activities. In the third process, which is called evaluation, evaluation of criteria is carried out, and alternatives are compared according to the criteria. In the fourth process, which is called aggregation or choice, evaluations are aggregated and a decision is made before planning for its implementation in the fifth process. This process is called implementation. A DSS should support one or more of these stages.

The activity diagram illustrates the activities of decision participants in the processes of decision making. The next section describes the object-oriented representation of the decision making procedure, which, in addition to the activity diagram, lays the foundation for the implementation of the DSS.

4.8 Object-Oriented Representation of the MCDM Decision Making Procedure for Natural Resource Management

The activity diagram in the preceding section illustrates the activities of decision participants and time sequence of decision making activities. This section uses the object-oriented concepts to represent the MCDM decision making procedure for natural resource management. Object orientation terminology was used in the last section to represent the general MCDM decision making procedure for natural resource management. It illustrated the general abstraction of various decision entities that need to take into consideration during the decision making processes. Representation of the decision making procedure with object-oriented concepts is presented in this section to lead to a fundamental object-oriented model for the MCDM decision making procedure. This modelling in turn can facilitate the analysis and design of DSSs since it can identify the most essential and fundamental classes and class relationships that the DSS implementation requires.

Notations to describe classes and their relationships are illustrated in Chapter 2. A class, as a set of objects with the same attributes, operations, relationships, and semantics, is rendered as a rectangle with the class name inside. Names, attributes, and operations of a class can be listed in the rectangle, separating with vertical lines. The class name is in the upper section in the rectangle while class attributes and operations are in the middle and lower sections. Operations usually end with a pair of brackets. Figure 4.11a shows a sample class of human being, which has some attributes, such as name, age, and weight, and some operations such as sleep, eat and read. For the purpose of legibility, attributes and operations are omitted from the rectangle sometimes even though they have been defined. The omitted form of the class of human being is shown in Figure 4.11b. A line connecting the two classes indicates an association showing a static general class relationship, usually an unspecified relationship. For example, a human being reads a book as shown in Figure 4.11b. A solid-headed arrow pointing from one class to another represents an inheritance relationship, the arrow pointing to the inherited class. Figure 4.11c shows the inheritance relationship between the class of book and the class of dictionary as dictionary inherits the properties of class book. A filled diamond indicates an aggregation of classes, the diamond side of the line as a composite class and the other side as member classes. Figure 4.11d shows the aggregation of class book as the composition of class cover and class content. Arrowhead lines represent flows of control or message, indicating an association as well. Figure 4.11e shows the proactive action of human being towards book.

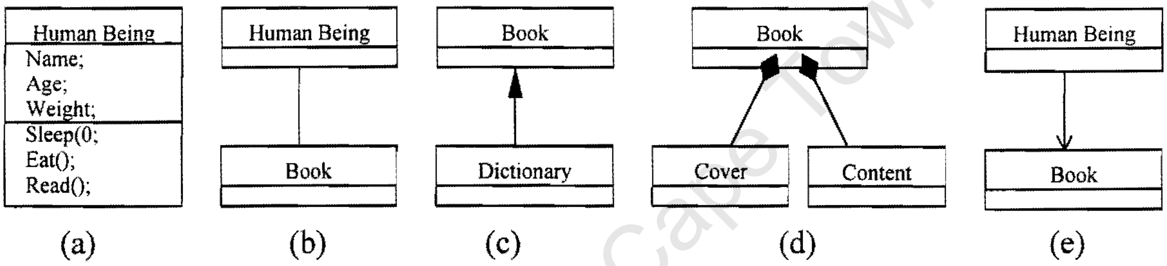


Figure 4.11: Class Notations

Figure 4.12 shows the class relationships of aggregation and inheritance among some fundamental classes for decision making activities. Some composite classes may include components that contain more specific information and some may be classified into several categories. Class Alternative, which represents a decision policy option, is composed of, as shown in (a), Class Action Element, which indicates an aspect of the action elements whose collective set constitutes the vector of attributes for the decision alternative. An object of Class Alternatives may contain several objects of Class Action Element, representing different action elements as an alternative. Figure (b) indicates that an alternative set is aggregated by several alternatives while Figure (c) shows a criterion hierarchy is constructed from some criteria. A criterion in turn may contain some features such as weights and value models, as shown in (d). Two general inheritance relationships are also shown in figure 4.12. Class Choice representing an evaluation result is extended into two specific classes, as shown in (e), of Overall Choice that represents an overall result of final evaluation, and Interest Choice that represents the evaluation result of individual decision participants. The two classes inherit the features from the parent class, Choice. Figure (f) shows that Class Overall Criterion Hierarchy, which represents the overall hierarchy of criteria that are associated with different interest groups, and Class Interest Criterion Hierarchy, which represents the criterion hierarchy of each interest group, are built based on Class Criterion Hierarchy with additional specific features.

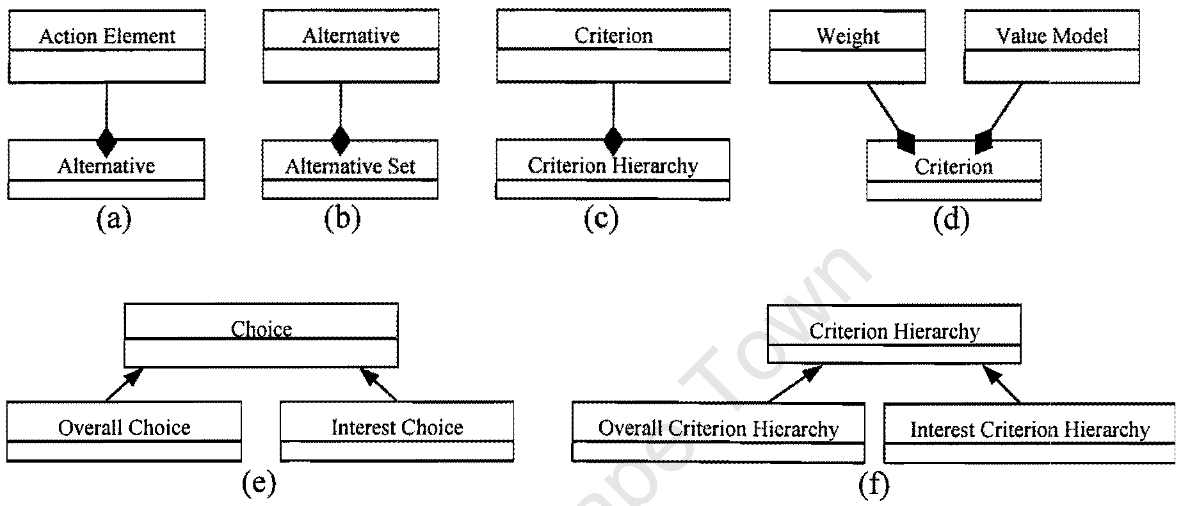


Figure 4.12: Class Components for MCDM Decision making Procedure

Figure 4.13 shows the general object-oriented representation of the MCDM decision making procedure for natural resource management problems. It is build based on the combination of Figure 4.9, which models the general MCDM decision making procedure, and Figure 4.12, which represents the class relationships, together with the activities of decision participants as shown in Figure 4.10. The object-oriented representation is simplified so as to keep the figure easy to be read and understood by avoiding sophistication and complexity in drawing the representation graph. Class operations are ignored, and are only indicated in the figure with a line within the class rectangles. In addition, different decision participants, i.e. domain experts, the facilitator, and stakeholders, are grouped as one entity in the figure although they are not assumed to play the same roles in the decision making processes. They are connected to the generalised classes such as Choice and Criterion Hierarchy instead of their sub-classes, which in fact are associated with individual participants during the decision making processes. For example, stakeholders are associated with classes of Interest Choice and Interest Criterion Hierarchy while domain experts and the facilitator are with all sub-classes of Choice and Criterion Hierarchy for the purpose of check and validation as well as their own decision making issues.

Classes in the figure interact with each other by sending and receiving messages through class relationships in order to execute various decision making processes. Decision participants, including stakeholders, domain experts and the facilitator, may first obtain perceptions on the problem from the problem environment; and brainstorm and identify various decision elements such as alternatives, action elements, criteria, and value models. Action elements are components of alternatives, as indicated in the figure with an aggregation relationship between Class Alternative and Class Action Element. Decision participants then generate the alternative set out of the alternatives identified, and this is also indicated in the figure with an aggregation relationship between Class Alternative Set and Class Alternative.

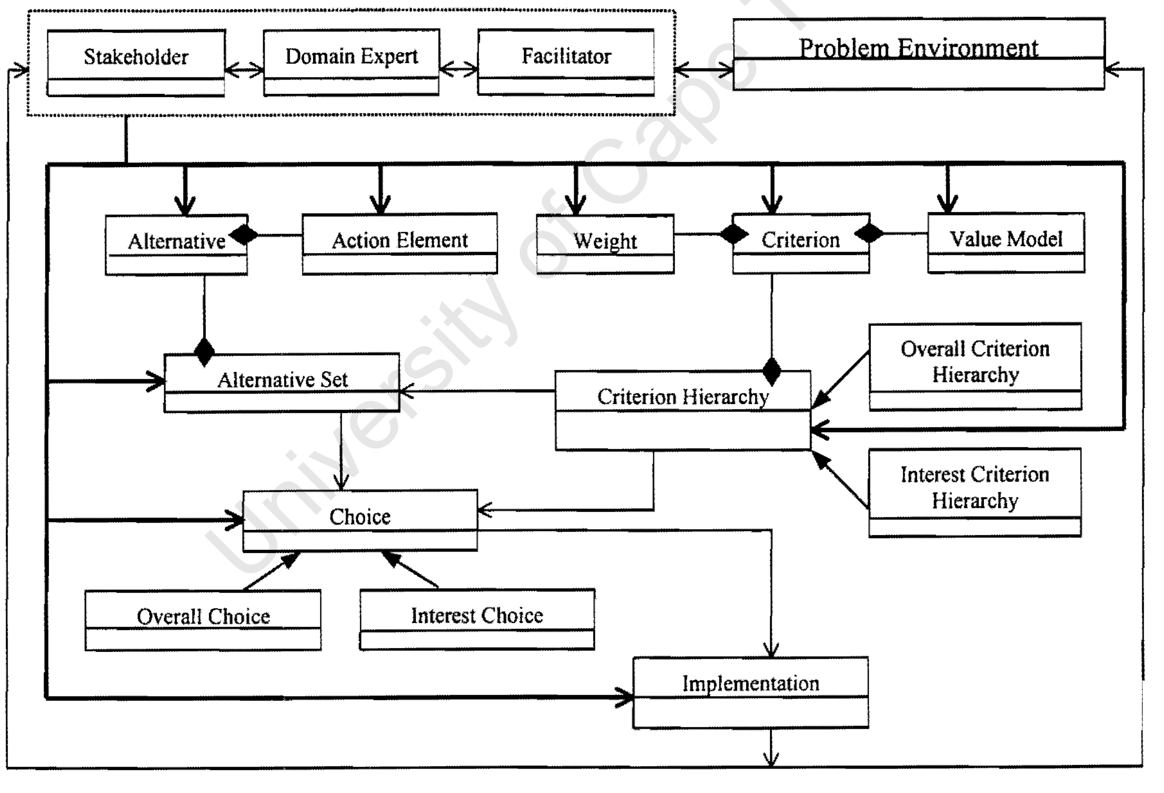


Figure 4.13: Object-oriented Representation of MCDM Decision making Procedure

Alternatives in the alternative set are evaluated according to the corresponding criterion hierarchy, indicated by a class association between Class Criterion Hierarchy and Class Alternative Set. Criterion hierarchies are constructed out of the criteria identified, as indicated in the figure with an aggregation relationship between Class Criterion Hierarchy and Class Criterion. The criteria in the corresponding criterion hierarchy are weighted and keep the weighting results in the weight class. The aggregation relationships between Class Criterion and two classes of Class Weight and Class Value Model are represented in the figure with a line with a filled diamond at the Class Criterion side. Finally the evaluation is aggregated to obtain a result, which is called Choice in the figure.

Implementation of the decision made may be planned and carried out by relevant agents with possible participation of decision participants. For example, a commercial consulting firm may take up the implementation planning mission after governmental organisations have accepted and granted the choice being made. Due to the implementation planning and other decision making activities, insights are obtained at various points about the problem and the problem environment. This may lead to further iteration of some decision making processes. Iterations and feedback of insights are indicated with directed lines pointing back to decision participants and Class Problem Environment.

Only general associations are represented in order to simplify the figure although there are in fact specific association between sub-classes. The general association between Class Criterion Hierarchy and Class Alternative Set is indicated with a line pointing from the former to the latter. Two associations, indicated by two lines pointing from Class Criterion Hierarchy and Class Alternative Set respectively to Class Choice, are required to obtain an aggregation of evaluation. These associations send messages between the associated classes. The association between Class Choice and Class Implementation passes messages to trigger the operations of decision implementation.

The figure does not attempt to probe the behaviour of the classes. It however implies their principle operations that are described above. The identification of the fundamental classes for the decision making procedure, the description of how classes interact during the decision making processes, and the analysis of their principle operations constitute foundations for the analysis and design of DSSs, which are the topics of the subsequent chapters.

4.9 Conclusions

A conceptual framework for a decision problem macro system has been presented in this chapter for analysing multi-criteria decision problems and for understanding problem analysis (structuring) and the decision making procedure in natural resource management. This framework provides the context and basis for developing DSSs to facilitate decision making for natural resource problems.

Decision problems in natural resource management were analysed in a systematic way. The role of the actors, which are physical objects that has influence on the decision problem under consideration, was given very important attention, as they are the information collectors, transmitters, and processors with different backgrounds, interests and conflicting goals. Actors are considered critical and the most important in decision making. Methods of actor identification were proposed. The decision environment is identified along with influence factors and other environment elements, which influence the decision problem under consideration. These factors have impact on the choice of decision alternatives, and in turn the decision to be made will have influence on them. The decision context that sets the boundaries of the decision problem along with its interfaces to the external world was then diagrammed to show the context of the problem and the problem environment. The interaction between various decision elements and entities, and the general strategy to solve decision problems were described. The overall framework of problem analysis in natural resource management was then described.

The general MCDM decision making procedure was modelled from different points of view including the interaction of participants, activity sequence and activity synchronisation. Based on the analysis of the macro decision analysis system, the MCDM decision making procedure for natural resource management problems was represented with object-oriented terminology. The decision making procedure was interpreted as interactive ongoing phases of various activities under different influence factors, representing the views of the different interest groups. The decision problems and decision making entities in the procedure were modelled in terms of "classes" in object orientation. Fundamental classes for the decision making procedure were identified together with their principle operations. Classes interaction during the decision making processes was described. The object-oriented representation therefore constitutes foundations for the analysis and design of DSSs for natural resource management decision problems.

The next chapter extracts the classes of various decision elements and primary DSS components after the capture of the general requirements of DSSs for MCDM in natural resource management.

Chapter 5

Modelling of DSS Requirements and Class Extraction of Decision Elements and DSS Components

5.1 Introduction

A DSS is a computer-based system that belongs to the macro system of decision problem analysis, which is dealt with in Chapter 4. A macro system dedicated to solve natural resource management problems may include a broad range of entities such as decision workshops, interviews, communication media, reports, other decision making techniques, models, analytic tools, and human beings such as facilitators, analysts, decision makers, domain experts, other stakeholders, etc. A DSS may comprise some of these entities to facilitate the decision making procedure. The modelling and the object-oriented representation of the MCDM decision making procedure in natural resource management, which are discussed in Chapter 4, constitute the foundation for the analysis and design of DSSs for natural resource management decision problems.

The objective of this chapter is to capture comprehensively the general system requirements for DSSs for MCDM natural resource management decision problems and to extract primary classes of decision elements for decision analysis and DSS modelling. These general system requirements and various classes are actually part of the DSS model to be built in this study since they capture some generic aspects of DSSs. The capture of DSS requirements is based on the understanding of the fundamental functions and the system environment of DSSs. Decision making processes are analysed and the system infrastructure is determined. System users and actors are identified and explored in the system context. In addition, DSS evaluation principles are applied to the requirement analysis to ensure that these principles are met by the DSS under development. General system requirements for DSSs are finally represented with use cases. Classes of decision elements of decision problems and other basic system components are therefore identified.

This chapter is organised as follows. Section 5.2 discusses the decision making stages of MCDM in natural resource management and the basic requirements for DSSs for natural resource management. In Section 5.3, the basic infrastructure for DSSs for natural resource management is determined as Internet based GDSS. Section 5.4 and 5.5 define the system users and actors along with the system context for DSSs. In Section 5.6, the main system functions of DSSs are capture resulting from the modelling of decision making in the preceding chapter and they are represented with some use cases. In Section 5.7, the DSS performance evaluation principles surveyed in Chapter 2 are examined to obtain additional system requirements for DSSs. In Section 5.8 the comprehensive system requirements are summarised as a collection of use cases. Section 5.9 identifies the classes of decision elements of decision problems and other basic system components. Conclusions are contained in Section 5.10.

5.2 Basic Requirements of DSSs for Natural Resource Management

From the analytical viewpoint, supporting the processes of decision making is central to DSSs even though they provide other functions such as communications. DSSs facilitate managerial decision making by providing tools, procedures, models and data that add structure to the decision making processes.

A DSS should support one or more of the stages of the decision making procedure, which are generally reviewed in Chapter 2. As discussed in Chapter 4, the decision making stages for MCDM in natural resource management are usually described as consisting of several distinct and iterative stages: problem analysis, problem structuring, evaluation, aggregation (choice) and implementation. At the stage of problem analysis, the problem is understood and various aspects of decision elements are brainstormed. At the stage of problem structuring, the problem is defined and structured in terms of criteria, alternatives, business rules and relationships, related data, uncertainties and dependencies that need better understanding for the problem under consideration, constraints that bound the alternatives, and causal relationships between key variables. Some ideas brainstormed from the problem analysis phase are organised into a value tree, in which the lower levels indicate the criteria used for evaluation and the higher levels may be either be considered as categories or as outcomes of the combined effects of the criteria beneath them. The primary interests and goals are sub-divided into operationally meaningful criteria.

Decision alternatives are defined after the identification of decision action elements. A potential set of these action elements describes a policy alternative, which does not need to be fully comprehensive but rather to be representative of a broad range of possibilities so that no parties feel marginalised. In most cases, the process will be iterative in that the initial set of policy alternatives may include a very wide range of options. As one or two dominant scenarios begin to emerge, the others will be discarded and these will be refined.

The evaluation and aggregation phases elicit subjective judgements, values for probabilities, value functions for evaluating alternatives, value weights for measuring the trade-offs amongst objectives (criteria) and risk preferences. They may also include the performance of any necessary statistical analyses, the conduct of logic-based reasoning, and the calculation of the expected achievement of each alternative. Decision participants assess the policy alternatives holistically from the point of view of their own interests. Alternatives are compared at different levels of goal achievement according to the value tree. The assessment of the consequences of each alternative may involve the use of a variety of models (e.g. economic models, hydrological models), use of expert groups, and other traditional methods of project assessment, such as environmental and social impact studies and cost benefit analyses. Relative "importance" weights are then assigned to criteria. Criteria within a category are first compared, and then the relative importance between the different categories is compared. Once preferences to alternatives and weights to all the criteria are assigned, the achievements can be aggregated at different levels of the hierarchy according to an achievement measure.

Finally, the sensitivity of the weighted achievement to key probabilities, weights, risk preference parameters, and critical variables has to be examined. The stage of implementation mainly concerns the planning of tasks to carry out the decision made. Critical activities and resources may need to be identified to ensure the implementation achieve the demanded benefits.

To summarise, problem analysis concerns with the problem understanding and brainstorming. Problem structuring includes alternative generation, criterion identification, and criterion hierarchy construction. During the evaluation stage, alternatives generated are assessed according the criteria in the value hierarchy, and criteria are weighted. The aggregation stage focuses on the selection of decision alternatives and sensitivity analysis. The implementation phase mainly deals with the planning for the chosen alternative.

Despite the different stages of the decision making process, the ultimate objective of DSSs is to evaluate complex alternatives in terms of values under uncertain circumstances. One of the main points is to gain insight into how the defined alternatives differ from one another and to generate suggestions for new and improved alternatives (Buede, 1996). The next section discusses the issue of the location of decision sites where DSS functions are carried out.

5.3 Natural Resource Management DSSs: Internet Based Integrated GDSS

Decision making in natural resource management involves various participants including multiple stakeholders, domain experts, and a facilitator, as discussed in the previous chapter. It is actually a problem of group decision making, which can be facilitated by a GDSS, which is discussed in Chapter 2.

The issue of the location of decision sites needs to be addressed for the analysis and design of GDSS. Usually at some stages of decision making, presentation facilities are needed in order to obtain the initiatives from different interest groups, and communication means are also necessary to support group decision analysis with geographically dispersed participants. For example, Huber (1984) proposes that workstations, a large public screen and a network are needed to enable group members to enter the system their own ideas. While this is viable, decision participants still have to meet together, which in most cases is difficult to be practical due to the constraints of time contributions from some stakeholders, as their time and patience are very scarce resources. In reality it is always a priority to take advantage of meetings of participants for the more important issues, such as problem understanding and strategic thinking, than simply using a software package to express themselves. This situation is much like the computation of background and foreground applications, in which foreground computations deal with the most urgent jobs while background applications with the less urgent and slow jobs even though both of them are equally important. In the decision making case, DSS usage is like a background application while other tasks like problem understanding may be like a foreground one.

With the development of networking technology and Internet communication, Internet access is becoming popular among ordinary people. Browsing the Internet to obtain world-wide information might be an instinctive skill to many people nowadays. Decision makers of a group decision making system can be connected remotely to the network using a Web browser or some other communication tools. They can have access to documents and other information in distributed databases, knowledge bases and other information systems via appropriate tools. DSSs that enable users to interact with each other in these ways are called network based DSSs. For natural resource management decision problems, a world wide network, which is Internet, is necessary since it can offers access to a DSS virtually all over the world while LAN (Local Area Network) and WAN (Wide Area Network) cannot. Therefore, network based DSSs for natural resource management is usually Internet Based.

One of the advantages of Internet DSSs is that they do not require extra physical facilities and additional communication means to support group decision analysis with geographically dispersed participants. Internet DSSs can simply run on any computer that is able to browse the Internet. Another advantage is that decision participants do not need to use the system simultaneously. They can use it when convenient and can avoid time conflicts in heavily scheduled days. In Internet DSSs, interactive information can appear in textual and graphical form in the web browsers. Internet communication technology helps achieve the most equitable overall benefit with the least cost to individuals, user-sectors, geographic regions, and international partners since the Internet has become the dominant method of information access. People are using universal clients such as Web browsers and email readers to connect to any system, from anywhere, and at any time.

5.4 Identification of System Users

For the domain analysis of DSSs, comprehensive system requirement specifications are needed in addition to the basic functional requirements and the rudimental system architecture presented in the two previous sections. The specification analysis starts from the identification of system users. Users are complex information processing units, as can be observed from different points of view. They may be not only the essential part of the decision making procedure, but also interactive and communicative media. Users are also decisive in the acceptance and implementation of a DSS.

Users that will interact with a DSS are identified. A user is a role played by a physical person or group of people who will use the system. Users of a DSS are therefore also actors, as discussed in Chapter 4, of the macro system of problem analysis, and may have impact on the problem analysis. Stakeholders, for instance, are users of a DSS for natural resource management, and they are of course critical players in the decision making processes.

Based on the identification of the actors for the macro system for the decision problem solution in Chapter 4 as well as past experiences, users for GDSS for natural resource management are identified as including a system administrator, a facilitator/analyst, stakeholders, domain experts, observers, and decision implementation agents. A system administrator is a real world person administrating the system. The role of a system administrator in an Internet based DSS is much like that of a LAN server administrator supporting network access of local users. Although its detailed functions may vary in different systems, the administrator need to register system users, monitor system and user status, and deal with routine system services such as starting and shutting down of the system.

A facilitator/analyst is a real world person facilitating the decision making processes in the system. The facilitator/analyst is expected to co-ordinate other decision participants and assist them with problem understanding, problem structuring, judgement expression, and possibly system usage. The facilitator/analyst should have the full knowledge of system functions and system usage skills. Other decision participants always resort to the help of the facilitator/analyst in cases of difficulties since the only visible system supervisor to them is the facilitator/analyst instead of the system administrator. They do not care about the existence of the latter, which is invisible to them. The facilitator/analyst therefore must keep contact with the system administrator in the management of system operations. In many cases, the facilitator/analyst may act as the system administrator.

Stakeholders are the real world people that are considered, by the decision makers, domain experts and the facilitator/analyst, as necessary to take part in the decision making processes in the system. They may include those people, or groups of people, who represent those affect or are affected by the problem in some way. They may need to use the system to express their perceptions on the problem and to indicate their preferences of alternatives of solution. Stakeholders are the main users of the system in that they are the main sources of information about the decision elements.

Domain experts are the real world people who have expertise in the problem domain. Other decision participants, including stakeholders and the facilitator/analyst, may obtain information and expert opinion from them about various issues. Examples are the area needed of a certain type of land in order to preserve a species of animal, the factors that contribute to the growth of the local economy, and the impacts of a proposed dam. On the other hand, domain experts may also need to use the system to help them gain information about the expert opinion that is given to and by other decision participants. In these cases, domain experts do not have an idea about the issue inquired but may find out by using the system as well as some other raw material.

Observers are the real world people who want to study the system or the problem concerned for various reasons. People might learn the system before they become actual decision participants. High-level personnel may hope to supervise their representative's decision making activities in the system while they do not want spend their own time in participating actual activities in the system. Governmental organisations and senior officials may need to check the decision making results from the system before they can make a final decision about the problem. In this sense, they are actually the decision makers for the problem concerned.

Decision implementation agents are real world people or organisations who are going to implement the decision or recommendation resulted from the use of a DSS. The main purpose for them to use the system is to plan the implementation for the decision made. Different agents may be involved and they may need to communicate with each other to find an efficient schedule of implementation. Before they start their implementation planning, these agents may need to understand the decision making processes that produce the resulted decision or recommendation to have a sense of decision validity, which might be difficult, if not possible, to be assessed. Therefore, they should be able to participate the decision making activities in the system as observers before they play as implementation agents.

In object orientation, individual users are represented as "objects" while a type of user is represented as a "class" in the system. It is useful to distinguish between the physical, real world objects and those surrogate objects in the system implementation, which represent them. The real world objects need to be modelled to understand the actions and activities the system is to perform, and the constraints to use the system. The surrogate objects usually have the same attributes or a few extra attributes to indicate the state of the objects from the system's point of view, but typically, they only have operations of accessing their attributes. Users identified are the real world objects while their represented concepts in the subsequent sections are surrogate objects.

Interactions between identified users and the system under consideration are analysed. These users, together with other objects, interact with a DSS in many ways. More users and other actors that have impact on or are impacted by the system will be identified through the analysis of the system context, which is discussed below.

5.5 The System Context and System Actors

A system actor is a role played by users or other physical objects that will interact with the system under consideration and have influence on the system when interacting with the system, for example, an external file or system.

The system context defines the environment in which a system resides. A context diagram graphically shows the boundaries of the system along with inputs and outputs to the system from different system actors. A context diagram is a high-level object message diagram in which the system and other issues are shown as objects and inputs and outputs are shown as "messages" to and from the objects.

System actors are identified for the DSS under consideration along with the definition of the system context. According the analysis results from the decision making modelling in Chapter 4, other actors apart from users identified for GDSS include the decision problem context and external systems.

The decision problem context is the environment for the problem under consideration, as discussed in the previous chapter. It contains the problem concerned, the elements of the problem environment and their interfaces to the problem. A decision problem needs not only to respond the business and technological issues of the problem itself but also to deal with constrains, risks and influences from outside. The environmental factors are categorised as four interacting environment classes, i.e., resources, uncertainties, rules, and influence factors. A DSS interacts with this problem environment to get information from and feedback to. The figure for the macro-system of solving decision problems in the previous chapter shows this relationship and also constitutes the basis to construct the DSS context.

External systems are other software and hardware systems that interact with a DSS. These systems may help the DSS to obtain the data and information it needs for analysis. They may also rely on the system for its output to gain insights about the problem. An Excel application, for example, is an external system that may provide spreadsheet data as data of preliminary decision alternatives for solving a decision problem. Another example of external systems is a presentation system, which may produce initial perceptions of the problem as input to the DSS under implementation, and may also demonstrate information to and from the system. External systems are important decision making facilitation tools complementary to the DSS.

The context diagram for GDSS for natural resource management is shown in Figure 5.1 with the DSS under consideration at the core of the context diagram and with the decision problem context in the background. Interactions between the DSS and various system actors are actually discussed above where different system actors are introduced. Data and information input to and output from the system are described in short terms along with directed arrows between actor rectangles and the shadowed system block. All actors are represented as classes in the form of singulars.

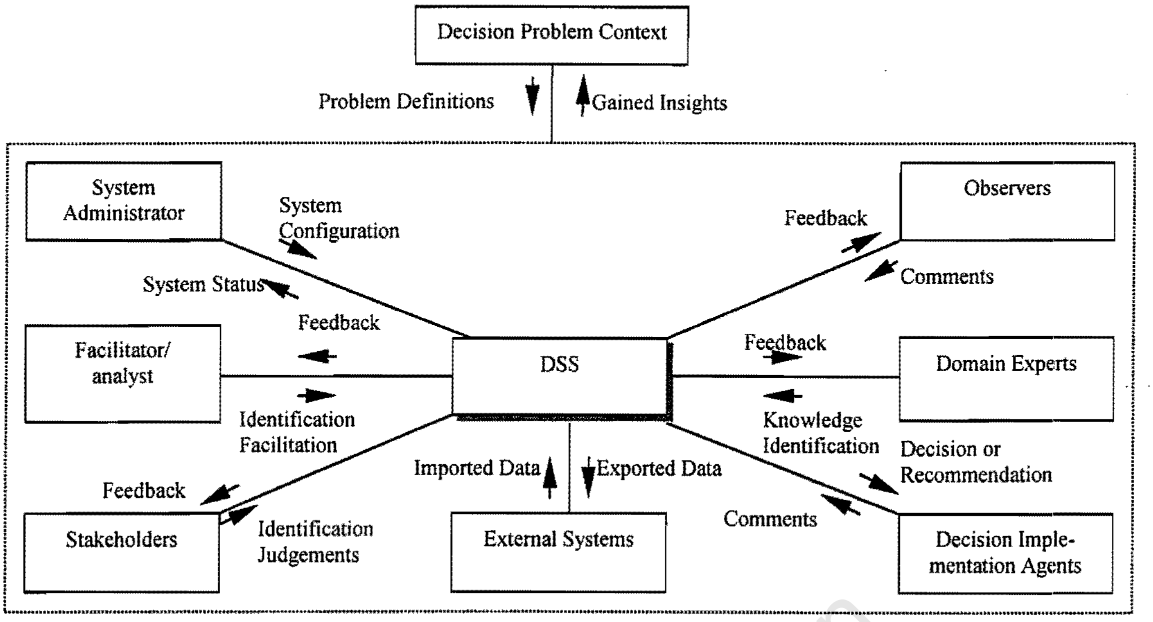


Figure 5.1: Context Diagram of the DSS

Various users are under the supervision of the system administrator who registers users and monitor the system status. Interactions between various users and the DSS generate the decision problem definitions, which might be based on the initial structure created by the facilitator/analyst by studying the decision problem preliminarily. Users communicate with each other via the Internet sharing information and data, which may be imported to and exported from external systems, or contributed by various users including domain experts. The facilitator/analyst needs to facilitate the stakeholders during the whole process of decision making. Observers can share various information and learn to use the system at the same time. Decision alternatives may be determined by the facilitator/analyst with references to the suggestions from stakeholders, and then evaluated and selected by using the judgements or preferences from stakeholders. The implementation then begins to be planned.

The identification of system actors along with the definition of the system context provides initial understanding of functions each actor performs in the system. This may lead to comprehensive system requirement analysis, which is described as system functions with use cases for each actor in the following sections.

5.6 Main System Functions: Use Cases and Case Descriptions

A use case (Jacobson, 1992; Martin, 1996; Gossain, 1998; Booch, Rumbaugh and Jacobson, 1999), which was first popularised by Jacobson (1992), is a short title for a system function to be performed by a DSS. It is usually connected with a system actor and contains a particular form or pattern of system usage by the system actor. A collection of use cases is thus actually a checklist of system functions. The concept of use cases is further illustrated by an example in the next subsection. The context diagram discussed in the previous section provides the best place to start identifying use cases by examining the interactions of each included actor with the system.

A case description is connected with a use case, and it is a brief description of what needs to be done in the system to implement the function included in the use case. A use case may have several case descriptions because of the variations in detailed system implementation it may have. A use case may include primary case descriptions that define essential implementation of the use case and secondary case descriptions that define its alternative implementation. A DSS might have a few dozen use-cases that capture its behaviour, and each use case might expand out to several case descriptions. To build a general model for DSSs, only primary case descriptions need to be included to define the main behaviour of DSSs since they can capture the generic system requirements of DSSs. Secondary case descriptions can be added at a later stage when it comes to detailed implementation for a specific system. System requirements are mainly described with use cases and case descriptions, which are the basis for further system analysis.

5.6.1 Primary Use Cases

Modelling the decision making procedure for decision problems of natural resource management in the previous chapter can lead to the development of primary use cases that capture the main requirements for DSSs in MCDM in natural resource management. Starting from the roles of different actors in the diagrams of general and MCDM decision making procedures, and the activity diagram, the primary use cases in Table 5.1 are identified for each actor. Some functions discussed at the modelling of the decision making procedure do not necessarily belong to DSSs. For example, the identification of stakeholders and domain experts is a function of the decision problem context instead of a DSS. For more information about the system functions included in the primary use cases in Table 5.1, refer to Chapter 4 where primary decision making activities were modelled.

Table 5.1: Primary Use Cases

System Actor and its Use Cases	Brief Description
Facilitator/analyst	
Guide for brainstorming of the problem	Guide for decision problem perceptions by stakeholders and domain experts
Guide for generation of alternatives	Guide for generation of alternatives by stakeholders and domain experts
Guide for construction of criterion hierarchies	Guide for construction of criterion hierarchies by stakeholders and domain experts
Co-ordination of decision making activities	Check and control of the progress of the decision making
Initialisation of relevant data	Initial data, information and structure
Generation of alternatives for evaluation	Selection of an overall alternative set for evaluation by stakeholders
Construction of the system level criterion hierarchy	Construction of an overall criterion hierarchy
Evaluation of overall criteria and final aggregation	Comparison of the overall criteria and final aggregation of all scores
Sensitivity Analysis	Sensitivity analysis of aggregation results
Stakeholders	
Brainstorming of the problem	Brainstorming problem perceptions
Construction of a criterion hierarchy	Construction of an individual criterion hierarchy
Generation or modification of alternatives	Generation/modification of individual alternatives
Evaluation of alternatives	Evaluation of the selected alternatives by the facilitator/analyst
Evaluation of criteria and individual aggregation	Comparison of the criteria and individual aggregation of scores
Sensitivity Analysis	Sensitivity analysis of aggregation results
Domain Experts	
Brainstorming of the problem	Brainstorming problem perceptions
Construction of a criterion hierarchy	Construction of an individual criterion hierarchy
Generation or modification of alternatives	Generation/modification of an individual alternative set
Evaluation of alternatives	Evaluation of defined alternatives
Evaluation of criteria and individual aggregation	Comparison of the criteria and individual aggregation of scores
Sensitivity Analysis	Sensitivity analysis of aggregation results
Check of decision elements brainstormed	Examination of decision elements brainstormed
Check of alternatives generated	Examination of generated alternatives
Check of the overall and individual interest criterion hierarchies	Examination of criterion hierarchies
Decision Implementation Agents	
Decision implementation	Implementation planning of the decision made

The concepts of use cases and case descriptions are further illustrated by an example below. A simplified form of a land use problem for exotic forest plantations (Stewart and Joubert, 1999) is used. The decision problem is about land use for plantations of exotic trees in a particular district. The principle stakeholders include a forest company, the national department of water affairs, the tourist bureau at the district, the national department of conservation, and the town council of the district. The forest company is mainly concerned about the profitability out of the timber produced by afforestation. Water supply will be affected if downstream river flows change. The tourist bureau cares about the tourist industry in the district while the conservation department is responsible for the environmental conservation. The town council mainly considers the personal well being for the residents in the area.

The example is given as an illustration to the concepts of use cases and case descriptions. As an example, a use case for the system actor of "stakeholder" is:

Use Case: Construction of a criterion hierarchy

A typical case description for this use case is:

Case Description:

Stakeholder enters the criteria hierarchy editor

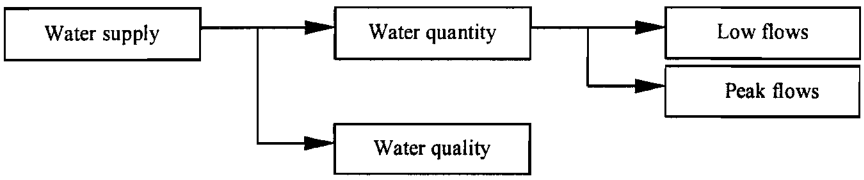
Stakeholder creates or opens a hierarchy

Stakeholder modifies a hierarchy by selecting a predefined criterion or by inputting a new criterion

Stakeholder saves the hierarchy

As an example to illustrate the use case and the case description, a DSS that supports the simplified land use problem may include a module that fulfils the requirements by the use case. The module may construct the criterion tree in the way that is generally stated in the case description. Suppose the stakeholder is "the national department of water affairs", who needs to use the system to create a criterion tree. The module is described briefly in the following box.

The stakeholder (system user) starts the module for criterion tree construction, and creates a new tree. After inputting all the criteria, the criterion tree looks like:



The stakeholder then saves the tree. At a later stage, the stakeholder may open (retrieve) the tree from a saved file and modify the tree by changing some criterion names, and deleting or adding some criteria. After the modification, the module should allow the stakeholder to save the tree for further analysis.

The use cases identified here provide a primary description of the system's basic functionality. Other use cases need to be discovered to describe the behaviour of the system more comprehensively.

5.6.2 Supplementary Use Cases

Further examination of the DSS context and the actors within it, the additional use cases in Table 5.2 can also be identified for some actors. The decision problem context inputs problem attributes to the DSS through facilitator/analysts, stakeholders, and other decision participants. It is no longer regarded as an actor due to its indirect interaction with the system. Use cases for the decision problem context can be embedded in those identified for other decision participants.

There are several supplementary use cases listed in Table 5.2. Two uses cases, i.e., "send messages' and "retrieve messages", are common among major system users, including the facilitator/analyst, stakeholders, domain experts, observers, and decision implementation agents. These two use cases provide a means of communication and co-ordination among system users. The use case for the observer indicates that observers may need to emulate the operations of a stakeholder, a facilitator/analyst, or a domain expert. An observer may be configured with typical privileges of a stakeholder, a facilitator/analyst, or a domain expert, and can carry out some typical functions of these decision participants. There are two use cases for external systems. A DSS may need to use data exported from an external system, and external systems may import data from the system for further processing. One additional use case is listed in the table for the facilitator/analyst. Users in the system may have different privileges of access to data, documents and other information for a specific problem case in the system. The facilitator/analyst can configure and modify these privileges for the users according to their functions, and possibly their demands. There are two use cases for the system administrator to administrate a system. For more information about these use cases, refer to Appendix A.

Table 5.2: Supplementary Use Cases

System Actor and its Use Cases	Brief Description
System Administrator	
Registration of users	User registration
Monitoring of system status	System status monitoring
Facilitator/analyst	
Configuration of user privileges	User privilege configuration
Observers	
Trial use of the system as a stakeholder, a facilitator/analyst, or a domain expert	Simulation of a stakeholder, a facilitator/analyst, or a domain expert
External Systems	
Exportation of data to the DSS	Data exportation
Importation of data from the DSS	Data importation
Facilitator/analyst, Stakeholders, Domain Experts, Observers, Decision Implementation Agents,	
Send messages	Message broadcast and mail
Retrieve messages	Message reception and reading

5.7 System Requirements and DSS Performance Evaluation Principles

As discussed in Chapter 2, it is very important to ensure the evaluation principles of system performance are met for a DSS since the success of a system is mainly assessed by using these principles. In this section, the evaluation principles are examined and use cases are discovered to meet the requirements for the system performance. Potential actors are also identified together with the use cases that an actor is associated with. DSS evaluation principles that are technical implementation oriented will be discussed later when DSS design is dealt with.

5.7.1 DSS Evaluation Principles and Use Cases

Table 5.3: DSS Performance Evaluation Principles and Use Cases

Evaluation Principle	Use Case	System Actor	Brief Description
Group Decision making Support	Support of multiple stakeholders	Stakeholder	Support of multiple stakeholders interacting with each other
Guidance in Decision making Processes	Decision making guidance	Facilitator/analyst, Stakeholder, Domain Expert	Availability of automatic information of decision making activities
Elicitation Techniques (Judgement Elicitation)	Judgement elicitation	Stakeholder, Domain Expert	Elicitation of textual and graphical values
Elicitation Techniques (Value Threshold Screening)	Value threshold setting	Domain Expert	Setting of value thresholds for action elements
Problem Analysis and structuring			
Problem Understanding	Problem orientation	Facilitator/analyst, Stakeholder, Domain Expert	Textual and graphical problem document display
	Previous case examination	Facilitator/analyst, Stakeholder, Domain Expert	Previous case demonstration
Brainstorming Techniques	Brainstorming of Decision Elements	Facilitator/analyst (as a guide), Stakeholder, Domain Expert	Brainstorming problem perceptions
Uncertainty Understanding and Handling	Uncertainty expression and management	Facilitator/analyst (as a guide), Stakeholder, Domain Expert	Expression and exploration of uncertainty during various processes
Alternative Creation	Creation of alternatives	Facilitator/analyst, Stakeholder, Domain Expert	Alternative generation
Value Structuring	Criterion hierarchy construction	Facilitator/analyst, Stakeholder, Domain Expert	Value hierarchy construction
Relationship Guidance	Expression of criterion relationships	Facilitator/analyst (as a guide), Stakeholder, Domain Expert	Exploration and expression of criterion relationship
Evaluation and Choice			
Results of Alternative Selection	Selection result presentation	Facilitator/analyst, Stakeholder, Domain Expert	Textual and graphical presentation of the selection made
Ways of Sensitivity Analysis	Sensitivity analysis	Facilitator/analyst, Stakeholder, Domain Expert	Sensitivity analysis on various variables
Model building	Decision model input	Facilitator/analyst, Domain Expert	Input of small and large decision models
Policy implementation Structuring	Decision implementation planning	Implementation Agent	Implementation planning of the choice
Reporting	Analysis result report	Facilitator/analyst, Stakeholder, domain expert, Implementation Agent	Textual and graphical report of the decision analysis result

Table 5.3 lists the DSS performance evaluation principles, the use cases designed to meet these principles, and the potential actors that are associated with each use case. High-level evaluation principles are in the bold font, and some of them contain further sub-principles.

A DSS for MCDM in natural resource management must support multiple stakeholders being involved in decision making for a specific problem, each of them representing a different interest group. Guidance in decision making processes is offered based on the information about the status of decision making. In addition to the guidance offered by the facilitator/analyst, the system may provide major decision participants with automatic information about the status of decision making activities of individuals and the system as a whole as well as tips of what and how to do next. This information may be retrieved from the database of progress trace.

Problem understanding is achieved via the display of textual and graphical documents and previous case demonstrations. Textual and graphical information stored in the database of cases and their documents can help understand the problem concerned. Previous problem cases of the same problem domain can be very useful since they allow users to learn from experience. A problem case is represented as an object with all its data and information (e.g. documents) as its attributes. A database of cases keeps these objects as previous experiences. Data and information of previous cases stored in the database of cases may be displayed upon request of system users. They are retrieved and displayed in textual or graphical ways.

Brainstorming techniques are used under the guidance of the facilitator/analyst to help stakeholders and domain experts perceive the problem concerned at the beginning of decision making processes. Also under the guidance of the facilitator/analyst, uncertainty is represented and is taken into consideration during various processes by stakeholder and domain experts. The first step of understanding and handling uncertainties is to identify uncertainties of various aspects. Exploratory approaches, which explore the possible outcomes or scenarios of uncertain issues, are usually used to study the uncertainties identified during the processes of decision making in a system. Decision alternatives of the decision problem are usually created through the joint effort of the facilitator/analyst and stakeholders. Criterion hierarchies are constructed. Domain experts may need to create their own alternatives and to construct their own criterion hierarchies for the purpose of carrying out their own decision making to obtain knowledge about some specific issues (refer to Chapter 4). Under the guidance of the facilitator/analyst, criterion relationships are expressed by and presented to stakeholder and domain experts. Relationships between criteria are useful in constructing criterion hierarchies. They are expressed when criteria are identified, and they are displayed when criterion hierarchies are constructed.

Judgements of the stakeholders about the decision problem and of domain experts related to their own specific decision making tasks should be elicited in ways that are convenient to system users. Textual and graphical ways are normally the most common methods of judgement indication. Value thresholds are set for action elements (alternative attributes) by domain experts. Value thresholds can screen out those alternatives that fall out of the value ranges for action elements. The system will then use these thresholds for alternative screening when an alternative set is generated. Sensitivity analysis for the decision problem may be carried out in many ways by the facilitator/analyst and stakeholders. The problem's main changing parameters, such as criterion weights, probabilities, and judgements of stakeholders, should be analysed in terms of the sensitivity of performance values of alternatives. Domain experts may also carry out sensitivity analysis for their own purpose of decision making.

Various decision models are input to the system to produce necessary data and information to help decision making. Facilities are provided in the system to allow the facilitator/analyst and possibly domain experts to build both small and large models into the model base. Models can then be modified afterwards. Stakeholders are usually not required to take part in model building. Results of alternative selection may be presented in one of the three forms, i.e., choice, sorted alternatives, and ranked alternatives. Textual and graphical approaches may be used for the display of these results. A report of the decision analysis result may be useful to the facilitator/analyst, stakeholders, domain experts, and implementation agents. Policy implementation structuring is carried out at the final stage of decision analysis. After the alternative selection, policy implementation is planned by implementation agents. A time schedule of action may be created.

5.7.2 Use Case Cross-Checking

Functions of some use cases identified resulting from the examination of the evaluation principles for DSSs may have been implemented by other use cases from the previous two phases of identification. By cross-checking all the use cases identified in each phase above, an aggregation of actors and use cases are obtained for the system as a whole. An examination of one by one is then made for the use cases derived from the DSS evaluation principles to screen out the unnecessarily repeated ones. Table 5.4 lists the use case designed to meet the DSS evaluation principles, the associated system actor, and the result of cross-checking, which shows if a use case is already implemented by previous use cases.

Table 5.4: Use Case Cross-Checking

Use Case	System Actor	Implemented
Support of multiple stakeholders	Stakeholder	Yes
Decision making guidance	Facilitator/analyst, Stakeholder, Domain Expert	No
Judgement elicitation	Stakeholder, Domain Expert	Yes
Value threshold setting	Domain Expert	No
Problem orientation	Facilitator/analyst, Stakeholder, Domain Expert	No
Previous case examination	Facilitator/analyst, Stakeholder, Domain Expert	No
Brainstorming of decision elements	Facilitator/analyst (as a guide), Stakeholder, Domain Expert	Yes
Uncertainty expression and management	Facilitator/analyst (as a guide), Stakeholder, Domain Expert	No
Creation of alternatives	Facilitator/analyst, Stakeholder, Domain Expert	Yes
Criterion hierarchy construction	Facilitator/analyst, Stakeholder, Domain Expert	Yes
Expression of criterion relationships	Facilitator/analyst (as a guide), Stakeholder, Domain Expert	No
Selection result presentation	Facilitator/analyst, Stakeholder, Domain Expert	Yes
Sensitivity analysis	Facilitator/analyst, Stakeholder, Domain Expert	Yes
Decision model input	Facilitator/analyst, Domain Expert	No
Decision implementation planning	Implementation Agent	Yes
Analysis result report	Facilitator/analyst, Stakeholder, Domain Expert, Implementation Agent	No

Functions included in some use cases in the table are already defined or implemented by the primary or supplementary use cases discussed above. Functions in some other use cases in the table can actually be considered detailed implementation of the primary or supplementary use cases. The use cases in the table that are already defined or implemented are marked with "Yes" in the status of implementation. The use cases in Table 5.4 that cannot be found in previously defined use cases are marked with "No" in the status of implementation. The use cases in the table that are not implemented previously are included as system requirements.

It is clear that the analysis of DSS evaluation principles can produce extra use cases, which are useful and important to the performance of a system but cannot be identified by using the conventional approaches. This justifies the importance and necessity of the usage of DSS evaluation principles in the DSS modelling.

5.8 Summary of System Requirements

The actors and the collection of use cases that capture the primary requirements for general DSSs are listed below in the tables of Table 5.5a-d, each representing a system actor and its use case set.

The identified use cases provide a primary description of the system's behaviour. They are typically at a high level but contain a sequence of events to carry out a particular use of the system. As seen from the processes of their generation, Table 5.5 includes the major use cases that are needed to meet the functionality and generic requirements of DSSs for MCDM in natural resource management.

Case descriptions are developed for each use case. Events and information in the case descriptions are focused on the system entities and their relationships. Case descriptions identified for natural resource management DSSs are shown in Appendix A. The case descriptions listed there are primary and generic to MCMD DSSs for natural resource management. Secondary case descriptions with exceptions or variations cannot be considered since they vary on different specific DSSs. For the modelling of generic system requirements, it is sufficient to look at the primary case descriptions for each use case since a comprehensive understanding of the desired general behaviour of the systems is obtained in this way.

Table 5.5a: Summary of System Requirements (Facilitator/analyst)

Facilitator/analyst & Use Cases	Brief Description
Configuration of user privileges	User privilege configuration
Send messages	Message broadcast and mail
Retrieve messages	Message reception and reading
Guide for brainstorming of the problem	Guide for decision problem perceptions by stakeholders and domain experts
Guide for generation of alternatives	Guide for generation of alternatives by stakeholders and domain experts
Guide for construction of criterion hierarchies	Guide for construction of criterion hierarchies by stakeholders and domain experts
Guide for uncertainty expression and management	Guide for uncertainty expression and management during various processes
Guide for expression of criterion relationships	Guide for criterion relationship expression and exploration
Co-ordination of decision making activities	Check and control of the progress of the decision making
Initialisation of relevant data	Initial data, information and structure
Generation of alternatives for evaluation	Selection of an overall alternative set for evaluation by stakeholders
Construction of the system level criterion hierarchy	Construction of an overall criterion hierarchy
Evaluation of overall criteria and final aggregation	Comparison of the overall criteria and final aggregation of all scores
Sensitivity Analysis	Sensitivity analysis of aggregation results
Decision model input	Input small and large decision models
Decision making guidance	Automatic information of decision making activities
Problem orientation	Textual and graphical problem document display
Previous case examination	Previous case demonstration
Analysis Result Report	Textual and graphical report of the decision analysis result

Table 5.5b: Summary of System Requirements (Stakeholder)

Stakeholder & Use Cases	Brief Description
Send messages	Message broadcast and mail
Retrieve messages	Message reception and reading
Brainstorming of the problem	Brainstorming problem perceptions
Construction of a criterion hierarchy	Construction of an individual criterion hierarchy
Generation/modification of alternatives	Generation/modification of individual alternatives
Evaluation of alternatives	Evaluation of the selected alternatives by the facilitator/analyst
Evaluation of criteria and individual aggregation	Comparison of the criteria and individual aggregation of scores
Sensitivity Analysis	Sensitivity analysis of aggregation results
Uncertainty expression and management	Expression and exploration of uncertainty during various processes
Expression of criterion relationships	Expression and exploration of criterion relationships
Decision making guidance	Automatic information of decision making activities
Problem orientation	Textual and graphical problem document display
Previous case examination	Previous case demonstration
Analysis Result Report	Textual and graphical report of the decision analysis result

Table 5.5c: Summary of System Requirements (Domain Expert)

Domain Expert & Use Cases	Brief Description
Send messages	Message broadcast and mail
Retrieve messages	Message reception and reading
Brainstorming of the problem	Brainstorming of problem perceptions
Construction of a criterion hierarchy	Construction of an individual criterion hierarchy
Generation/modification of alternatives	Generation/modification of an individual alternative set
Evaluation of alternatives	Evaluation of defined alternatives
Evaluation of criteria and individual aggregation	Comparison of the criteria and individual aggregation of scores
Sensitivity Analysis	Sensitivity analysis of aggregation results
Check of decision elements brainstormed	Examination of decision elements brainstormed
Check of alternatives generated	Examination of generated alternatives
Check of the overall and individual interest criterion hierarchies	Examination of criterion hierarchies
Uncertainty expression and management	Expression and exploration of uncertainty during various processes
Expression of criterion relationships	Expression and exploration of criterion relationship
Value threshold setting	Setting of value thresholds for action elements
Decision model input	Input of small and large decision models
Decision making guidance	Automatic information of decision making activities
Problem orientation	Textual and graphical problem document display
Previous case examination	Previous case demonstration
Analysis Result Report	Textual and graphical report of the decision analysis result

Table 5.5d: Summary of System Requirements (Other System Actors)

Actor & Use Cases	Brief Description
System Administrator	
Registration of users	User registration
Monitoring of system status	System status monitoring
Implementation Agents	
Send messages	Message broadcast and mail
Retrieve messages	Message reception and reading
Decision implementation	Implementation planning of the decision made
Analysis Result Report	Textual and graphical report of the decision analysis result
Observers	
Send messages	Message broadcast and mail
Retrieve messages	Message reception and reading
Trial use of the system as a stakeholder, a facilitator/analyst, or a domain expert	Simulation of a stakeholder, a facilitator/analyst, or a domain expert
External Systems	
Exportation of data to the DSS	Data exportation
Importation of data from the DSS	Data importation

5.9 Class Extraction of Decision Elements and System Components

After the system requirements are captured, it is then possible to extract the basic system building material out of the requirement specifications. This basic system building material includes decision elements of the decision problem and other generic system components, which are represented with classes of object orientation for the purpose of further exploration of decision analysis and DSS development.

A class is a collective representation of objects with the same attributes. After the examination of the use cases and case descriptions, the primary classes of decision elements and other system components for a DSS in MCDM in natural resource management are identified together with their definitions. These classes are called primary because more system and platform specific classes, which may be difficult to be generalised into generic elements for the DSS model, will be found at the later stages of system analysis and design. Class definitions document classes' roles and their responsibilities. They are also added to the class specification, which is part of the DSS model. The DSS model is actually the sum of all system modelling elements and their specifications. Additional information about the elements is entered into the DSS model as analysis proceeds.

The primary classes identified according to the requirements of DSSs are divided into two categories. One category includes the classes of decision elements. Another category contains the classes of other general fundamental system building components. More system and platform specific building classes will be identified as the process of system analysis and design comes closer to the actual implementation and coding of a specific DSS. Classes of decision elements and other system components are listed below in two divisions. Singular nouns with an initial capital letter and no space between words are used for class names.

Decision element classes are abstracted representations of generic decision elements for decision problems and DSSs in MCDM in natural resource management. They are collective sets of decision elements of the same features. Decision elements are the entities that intrinsically exist in a decision problem and the decision making processes. They usually form the basis of decision analysis and pay a definition for the decision problem under consideration. Decision element classes of decision problems and DSSs for MCDM in natural resource management are listed in Appendix B-1 together with their definitions.

Super classes for system users are created resulting from the commonality between various system user classes. The user classes are abstraction of real people or organisations, and contain some attributes in common, such as name, pass word, organisation, phone number, email address, post address, and physical address. Super classes of "user" and "DecisionParticipant" are created to hold the common structure, while unique attributes for each class remain in the individual classes. The definitions of two classes are included in the appendix.

Classes of DSS building components are the material used by system developers to construct a DSS. There are many kinds of system components. For example, system functions and procedures were the basic system components in the program systems prior to the object-oriented times. Classes are regarded as the most fundamental units in object-oriented systems. Primary system components of DSSs include the classes of decision elements defined above and other basic system building classes, which are listed in Appendix B-2 together with their definitions. Additional system component classes can be found when a specific DSS is developed with a programming language in a certain platform environment.

Classes should be defined along with their operations and attributes. The behaviours of a class are described by the operations of the class. Such operations are the domain-specific changes in state that the class may endure or the behaviour and responsibilities that it may be asked to perform. Potential operations of classes can be determined by examining the problem statement, use cases, class definitions, and other documents. Use case and case descriptions may include primary operations. Case descriptions describe actors' typical workflow. But activities in the workflow may contain those such as discussion with a fellow group member, browsing a manual, making a phone call, or performing other activities (decision element checking by domain experts, for example) that are part of the work but may not covered by the system. Case descriptions may contain such parts. Classes, however, do not since that they include only parts that will be implemented into a system that is designed based on the DSS model.

When modelling DSSs, attention is mainly focused on primary class operations (In the same way that we are only interested in some of the infinity of classes which might be perceived when observing a system). Minor operations, such as access operations used to get or set class attributes or associations, are not considered during the DSS model analysis unless they have significant and interesting domain meaning. This is because these operations are common to all the classes in the system. For instance, all attributes are assumed to have appropriate access operations. Moreover, some of the operations need to be refined because of the specific techniques people might use in their decision making processes. This can be done at a later stage or at the system design phase, depending the particular methods used to deal with different aspects of decision making issues, such as uncertainty analysis and evaluation approaches.

Class attributes are inherent properties or characteristics of the class. When a class receives a message it has direct access to its own attributes and the data passed with the message. Some attributes themselves are in fact classes that are defined in the same model. Literally every attribute is a class. For example, the “name” in the class "user" can be actually regarded as a class. It can perform some operations such as “create”, and has some attributes of its own. This, however, is only meaningful from the programmer’s point of view and can be captured later in the system design phase or in the programming language design. In the DSS model, only those classes are considered that are meaningful from a developer’s point of view and that can lead profound insights into the DSS model. The attributes and operations for the classes of decision elements and system components are listed in Appendix C.

5.10 Conclusions

This chapter has conceptualised in detail a general statement of requirements for DSSs for MCDM in natural resource management. The primary purpose of this general statement has been to provide a set of core requirements and behaviour for a DSS. This requirement set and the classes captured out of it are actually part of the DSS model to be built in this study since they are some generic aspects of DSSs.

Requirements were determined along with the analysis of system actors who interact with the system. Actors are regarded as the critical and most important elements to take into consideration in a system. The system context helps the identification of actors and the interaction between actors by setting up the environment and boundaries of the system.

Requirement specification was also facilitated by the analysis of DSS evaluation principles. This will serve as a purpose of ensuring the DSS evaluation principles met by the potential system to be implemented. Generic system requirements were then described with system behaviour and represented as a collection of primary use cases and case descriptions. As a result, classes of decision elements of decision problems and other basic system components were identified.

The basic system building material, which includes decision elements of the decision problem and other generic system components, was captured out of the requirement specifications. These decision elements and DSS components were modelled with classes of object orientation for the purpose of further analysis of decision making and the development of DSSs. The next chapter discusses the roles of these classes in MCDM.

Chapter 6

Roles of Classes of Decision Elements and DSS Components in MCDM

6.1 Introduction

After the identification of decision element classes of decision problems and some primary system component classes of DSSs for MCDM in natural resource management, it is very important to further analyse the classes of these system entities, especially decision element classes as they can be also a basic representation form of problem domain knowledge in decision analysis as discussed in Chapter 3 and 4. On the other hand, a DSS is composed of system building components and has to deal with decision elements of the problem concerned, which are a kind of system building material. In fact both system components and decision elements are part of a DSS.

One main objective of this chapter is to analyse the roles of the classes of decision elements and primary DSS components and class relationships in decision analysis and DSS development. At the same time this chapter builds a general class framework for decision analysis and DSSs for MCDM in natural resource management. Decision elements and system components are represented as classes in object orientation. These classes are derived from the modelling of the problem context and the decision making procedure (see chapter 4) and the specification of DSS system requirements (see chapter 5). They are modelled together with their attributes and operations as well as the analysis of the class interaction among the classes, which also describes the system behaviour of DSSs. The object-oriented representation of these classes together with the relationships of associations, aggregations and inheritance between the classes illustrates decision analysis in an object-oriented way.

Decision classes and their relationships are very important in many aspects in decision analysis and DSS development. Classes and class interactions represent the main activities of decision analysis as well as the major functions of a DSS. The behaviour of classes is a set of operations and mainly represents various decision making actions. Reusable classes are the basis for a specific decision problem to be structured and for a DSS to be developed. Various classes and class interaction diagrams offer different hierarchies and points of view for decision problems and DSSs, and provide an easy way to bridge the gap between decision analysts, stakeholders, domain experts, and DSS researcher. Class hierarchies provide a mechanism to manage complexity in decision analysis and DSS development. Especially, decision classes and class interactions are very important in determining the resources and paths for decision making. The analysis of dynamic decision classes offers the internal state changes of some important decision entities during various phases of decision making and may demonstrate some group decision making paths by major decision participants for a decision problem. Class relationships are also useful in decision analysis and DSS development. They are able to offer a mechanism to construct specific decision entities for a decision problem. They can provide a hierarchical or composite point of view for some important parts of a decision problem and of a DSS. They can also to mange complexity and reuse existing proven knowledge and past experiences, leading to well understanding of the problem and high productivity and high quality of decision analysis and DSS implementation.

By integrating these classes and various class relationships into a single diagram, a general class framework for decision analysis and the development of DSSs for MCDM in natural resource management is then obtained. The functions of decision analysis and the behaviour of DSSs can be integrally explored in this way.

This chapter is organised as follows. Section 6.2 discusses roles of classes and class interactions in decision making. Class interaction diagrams are used to show the static and dynamic decision making behaviour. Section 6.3 demonstrates resources and paths for decision making based on classes and class interactions. Roles of dynamic classes for decision making are further analysed in Section 6.4 with the state transition analysis of the most significantly dynamic classes in the classes of decision elements and system components. Roles of class relationships are analysed in Section 6.5. A general system framework composed of classes and their relationships is then presented in Section 6.6. Conclusions are contained in Section 6.7.

6.2 Roles of Classes and Class Interactions in Decision making

Classes are the abstractions of the individual objects with the same properties. Decision elements and other primary DSS components are generalised as classes to model decision analysis in the study. Classes define the entities, either physical or non-physical, which need to be examined during the decision making procedure, while class interactions model the dynamic aspects of decision making, defining the activities and action paths to take for decision analysis. Class interaction diagrams offer a means to show the entities and paths for decision making.

6.2.1 Class Interaction Diagrams

Classes and their interactions can be graphically represented by class interaction diagrams for decision analysis and DSSs alike. In class interaction diagrams, each class is drawn as a rectangle, and the messages are shown as text ended with brackets beside the connecting lines between the classes as shown in Appendix D. Naming of behaviours and attributes will follow this style: starting with a lowercase letter, no space between words and the first letter of each additional word being capital (e.g. roomNumber).

Class interaction diagrams are primarily constructed around actors of DSSs on the basis of one diagram for one actor. This is a new approach introduced here for drawing class interaction diagrams. It is called *actor-oriented class interaction diagramming* and can cater to the modelling of MCDM DSSs, which usually have more actors than other kinds of systems.

This approach has some advantages over the traditional methods, in which one or more class interaction diagrams are drawn for each use case of a system. First, the approach can produce integrated diagrams while showing the class interactions and grouping the interacting classes together for the same actor. All the case cases for a certain actor are integrated into a single diagram to obtain a comprehensive point of view of the roles the actor plays in the system. Secondly, there is no need to patch them up across several diagrams as is normally done in most object-oriented methods in which each use case is modelled by one or more interaction diagrams depending on its complexity. Thirdly, different points of view are also obtained from critical classes such as actors for the DSS model as each actor has its own class diagram to show the interaction between classes. Further analysis will then be concentrated on those classes with significant dynamic behaviour.

The one disadvantage for the actor-oriented class interaction diagramming is that the interaction diagrams constructed for some actors in the system may be too complicated to read. In this case it is suggested that a diagram be divided according to the behaviour of the actor involved, extra graphical notations be used, or those class interactions that are common to some actors are separated as sub-diagrams. Class interaction diagrams for all system actors are shown in Appendix D. It is noted that only primary class interactions are shown in the diagrams. Dashed lines are used when lines have to cross over each other in order to make the diagrams easy to read.

Class interaction diagrams of the primary use cases of the major decision participants, i.e., the facilitator/analyst, stakeholders, domain experts, and decision implementation agents, are useful for the modelling of decision analysis and DSSs in an object-oriented way. The major functions of decision analysis and also of a DSS are explored graphically in this way. Class interaction diagrams can analyse class behaviours, which are in fact various decision making activities. They can clearly show the decision making paths for various decision participants and the interactions of various decision elements and other primary system components. Interaction diagrams also model the interactions and flow of control that characterises the behaviour of a DSS as a whole, including use cases, patterns of system behaviour, mechanisms of communications, frameworks of class interactions, and the behaviour of system components such as decision elements.

6.2.2 Roles of Classes and Class Interactions

In addition to their essential roles in the analysis and design of DSSs, classes and class interactions contribute to modelling for decision analysis. For example, the four interacting classes: resources, uncertainties, rules, and influence factors, as discussed in Chapter 4 for the problem environment of MCDM in natural resource management, provide the basic sources of decision alternatives based on perceptions of actors towards their subjective view of the problem. Besides their roles in determining the resources and paths for decision making and as a fundamental way to represent various kinds of knowledge (see Chapter 3 and 4) for decision making, classes and their interactions play many other important roles in decision making. Some of the more obvious ones are listed below.

- Classes and class interactions included in class interaction diagrams offer a mechanism to investigate various classes involved in decision making and to define decision making paths including decision making activities by investigating the actual interactions of various classes of decision elements and other primary DSS system components.
- Well-identified reusable classes that have been tested in the field on similar decision problems may become the basis for a specific decision problem to be structured and for a DSS still largely to be developed.
- Classes and class interaction diagrams offer an easy way to bridge the gap between decision analysts, stakeholders, domain experts, DSS researchers and developers. Classes and their graphical representations can help to make decision problems, decision making and DSS development more easily understandable to different decision participants and to DSS personnel alike.
- Classes and class hierarchies offer a mechanism to manage complexity in decision analysis and DSS development. Class inheritance, in which a child class inherits the properties of its parent classes, and class abstraction of the key elements of the problem domain may simplify decision analysis by reducing the number of independent decision elements that need to be identified at the first phase.
- Classes and class interaction diagrams provide different hierarchies and points of view for decision problems and DSSs. Each class (or object in a specific case) and their clusters can represent a different part of the problem and the DSS under consideration, offering different points of view for the problem, the decision making paths, and the development of the system.
- Class interaction diagrams provide an efficient way to identify major class operations, which represent various decision making actions and are an essential part of classes, by considering possible messages that one class may pass to another. In fact, class interactions are explicitly labelled by using class operations as interaction names together with control flows represented by arrows, as shown in the class interaction diagrams in Appendix D. Appendix C lists the operations for classes of decision element and system components as well as their attributes.

- Classes and class interactions may be used to represent both the main activities of decision analysis and the major functions of a DSS.

As an instance of the roles of classes and class interactions, the role in determining decision making resources and paths by classes and class interaction is demonstrated in the following section by using class interaction diagrams for the major decision participants.

6.3 Resources and Paths for Decision making

Classes of decision elements and other primary DSS components, and class interactions provide a convenient way to determine the resources and paths for decision making for different decision participants. Whereas classes and their relationships model the static aspects of decision making, defining the entities, either physical or non-physical, to look at when considering various constraints, values and decision opportunities, class interactions model the dynamic aspects of decision making, defining the activities and action paths to take during the decision making procedure.

6.3.1 Resources and Paths of Decision making for the Facilitator/analyst

Figure 6.1 is the simplified class interaction diagram for the facilitator/analyst constructed on the basis of the identified classes. Relevant important classes are all included with indications of control flows between classes. Individual function names of class interactions are omitted from the diagram for the purpose of conciseness. For the detailed class interaction diagram, refer to Appendix D.

As shown in the class interaction diagram, the main entities for the decision making of the facilitator/analyst fall into four class hierarchies, i.e., communication, problem structuring support, problem structuring, and evaluation. The class hierarchy of communication includes the classes of main decision participants, i.e., the classes of stake holders, implement agents, observers, and domain experts, the class of messages, and the class of progress trace. Definitions of these classes can be found in Appendix B. The class hierarchy of problem structuring support includes the classes of case documents, uncertainty, uncertainty dependency, event probability, criterion relationships, alternative constraints, action rules, decision models, and attribute value thresholds. The class hierarchy of problem structuring consists of the classes of decision attributes, decision alternatives, system criteria, decision alternative sets, system alternative sets, criterion hierarchies, and system criterion hierarchies. The class hierarchy of evaluation includes the classes of judgement, evaluation data, achievement measures, evaluation results, and system evaluation results. These four class hierarchies are indicated in the diagram.

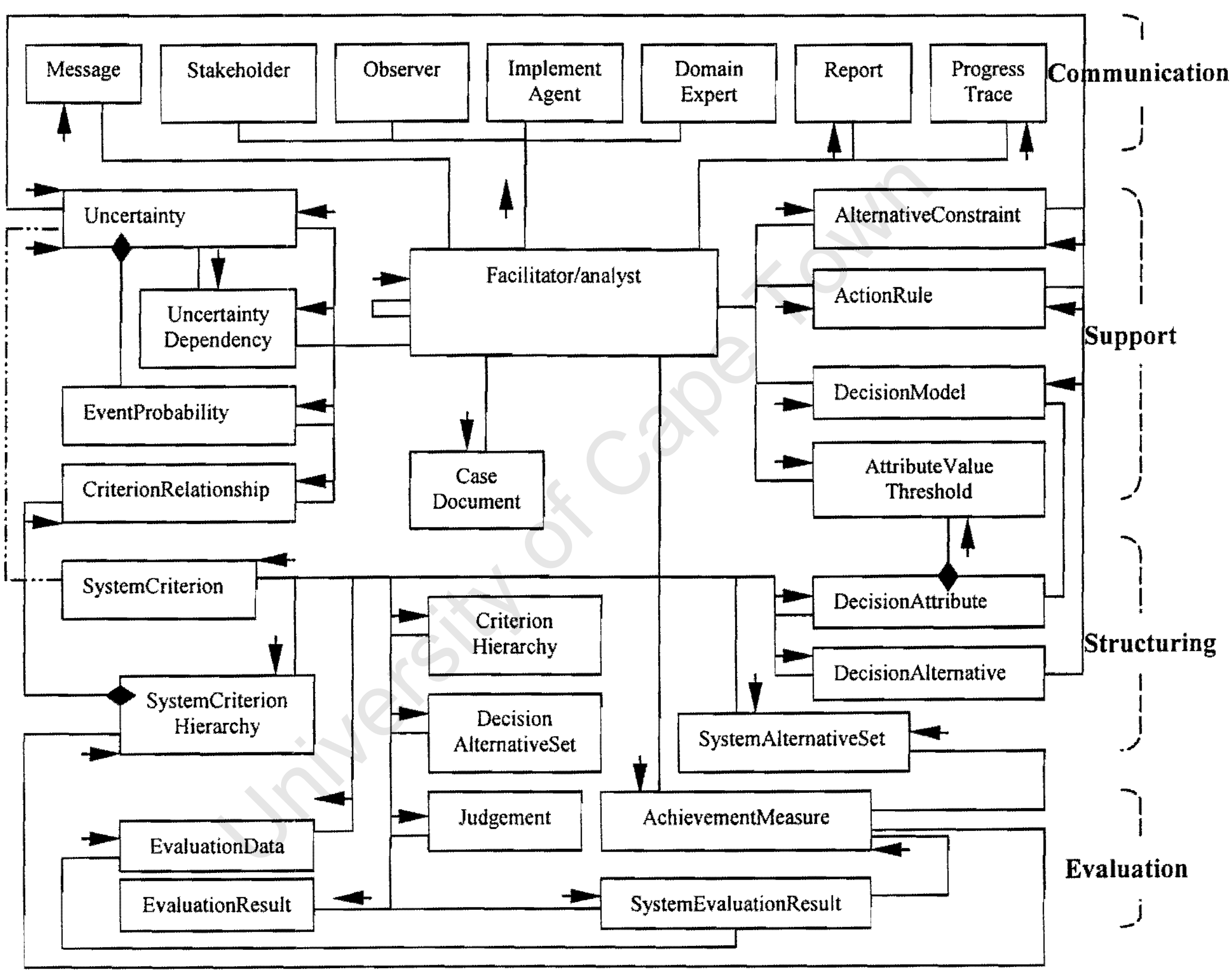


Figure 6.1: Simplified Class Interaction Diagram for the Facilitator/analyst

Resources of decision making for the facilitator/analyst are determined by the entities represented in the class interaction diagram. The specific mechanism and constraints of communications between the facilitator/analyst and other decision participants are considered after examining the individual objects, which are the class instances for a specific decision problem, included in the class hierarchy of communication. Various aspects of individual objects (e.g. documents of past cases) defined in the class hierarchy of problem structuring support are examined to decide other sources of decision opportunities such as uncertainty areas and decision alternative constraints, and resources of decision making such as action rules and value thresholds of decision action attributes. On the basis of the entities included in the class hierarchy of problem structuring, the problem under consideration is structured. Decision alternatives and criteria are evaluated by using various means required by the entities in the class hierarchy of evaluation.

At the same time, paths of decision making are determined by the class interactions, which are indicated in the diagram by arrows and involve many decision making activities. From the overall point of view, the four class hierarchies demonstrated in the diagram reflect the decision making processes discussed in Chapter 2 and Chapter 4. Decision making starts from communication among decision participants. Various support materials are collected afterwards to appropriately structure the decision problem under consideration. Evaluation is then carried out.

On the other hand, class interactions included in each class hierarchy and among the four class hierarchies reveal detailed decision making activities and paths. This section is not intending to describe all the detailed decision making activities and paths but to demonstrate how class interactions can determine them by showing some main decision making paths. The class interactions included in the class hierarchy of communication indicate the path of co-ordination of various decision making activities among various decision participants by the facilitator/analyst. From the very beginning to the final moment of the decision making procedure, the facilitator/analyst needs to check the decision making progress of individuals, inform various parties concerned, and initiate new transactions to carry out the decision making tasks. The role of the facilitator/analyst is mainly that of facilitation and co-ordination in the four class hierarchies.

In the class hierarchy of problem structuring support, the facilitator/analyst guides the exploration of various aspects of the decision problem by the decision participants, whose results can be used to analyse uncertainties and to validate decision action attributes, criteria and decision alternatives. The class interactions show this kind of automatic or non-automatic validation and value examination by a DSS, which may not be part of decision making activities of the facilitator/analyst. Values of decision attributes are screened with attribute value thresholds and may be calculated with decision models to obtain derivative information. Criterion relationships are useful in constructing system level criterion hierarchies. Decision alternatives are examined with action rules and alternative constraints, which need to be considered under uncertainties. Event probabilities may be part of uncertainty. Uncertainties may also be taken into consideration for the construction of system level criteria. Uncertainties may be explored with various dependencies among uncertainty elements.

In the class hierarchy of problem structuring, decision elements of decision alternatives and system criteria are organised in the form of alternative sets and criterion hierarchies respectively. Criterion hierarchies and decision alternatives generated by the decision participants under the guidance of the facilitator/analyst are checked before the evaluation is carried out.

In the evaluation hierarchy, judgements about the relative importance of system level criteria and final aggregation are made. Evaluation data and perhaps evaluation results resulted from the evaluation and score aggregation by stakeholders are checked as well as system level criterion hierarchies before the final score aggregation is calculated with achievement measures to obtain system level evaluation results.

It is obvious that there are control flows in the diagram between the four class hierarchies. Decision making activities included in the class hierarchy of communication offer the basic means for decision participants to communicate, leading to the identification of support material for problem structuring. Decision problems are constructed based on the elements and various considerations identified by the decision making activities indicated by the class hierarchy of problem structuring support. The evaluation of decisions is based on the criterion hierarchies and the alternative sets defined during the problem structuring.

6.3.2 Resources and Paths of Decision making for Stakeholders

The simplified class interaction diagram for stakeholders is shown in Figure 6.2 in the same way as that for the facilitator/analyst. For the detailed class interaction diagram, refer to Appendix D.

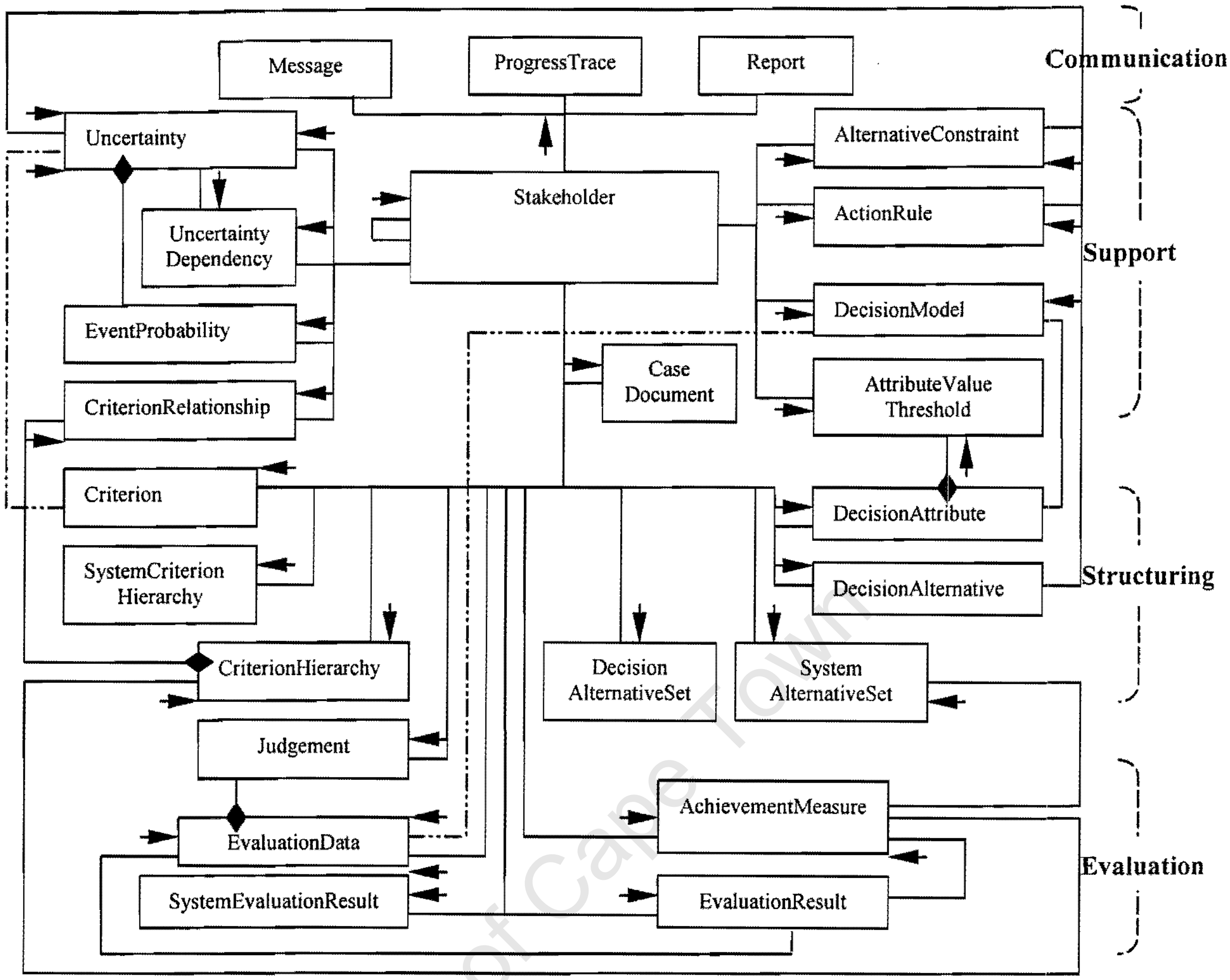


Figure 6.2: Simplified Class Interaction Diagram for Stakeholders

There are many similar aspects between the analysis of the resources and paths of decision making for stakeholders and that for the facilitator/analyst. Firstly, as indicated in the diagram, for the decision making procedure for stakeholders, there are the same four class hierarchies, some including more or less the same entities, as for the facilitator/analyst. For both stakeholders and the facilitator/analyst, there are the same general interactive decision making processes, including communication, decision structuring support, problem structuring and evaluation. Secondly, like those for the facilitator/analyst, resources of decision making for stakeholders and various sources of decision opportunities are determined by the entities represented in the class interaction diagram. The main communication mechanism used by stakeholders, for example, is via messages and via tracing decision making progress. Thirdly, like those for the facilitator/analyst, class interactions for stakeholders included in Figure 6.2 reveal detailed decision making activities and paths, some of which are quite similar to those for the facilitator/analyst. For example, the decision making activities involved in the second class hierarchy, which can automatically carried out by a DSS or manually by stakeholders, are almost the same as those discussed for the facilitator/analyst except that stakeholders focus on individual decision criteria while the facilitator/analyst is mainly concerned with system level criteria.

The similarity in the entities that the facilitator/analyst and stakeholders have to deal with in decision analysis reveals some useful insights. The classes included in the class hierarchies of problem structuring support and evaluation are exactly the same for both the facilitator/analyst and stakeholders. This is due to two reasons. The first is that the support phase is so fundamental in problem identification and problem structuring that it needs to take every possible aspect into consideration by all decision participants. The second reason is that the evaluation phase allows every main decision participant to find out how the decision result is obtained and to examine relevant values. Unlike that for the facilitator/analyst, the class hierarchy of communication for stakeholders does not include the classes of decision participants. This is because with the help of a DSS, stakeholders may not need to deal with other decision participants as the facilitator/analysts does during the decision making procedure. The facilitator/analyst may need personally to contact other decision participants and physically to co-ordinate their use of a system while stakeholders may only communicate with other decision participants virtually by using the system. There is also a difference between the class hierarchy of structuring for the facilitator/analyst and that for stakeholders. The class hierarchy for the facilitator/analyst contains the class of system criteria while that for stakeholders contains the class of criteria. This is because, during the process of problem structuring, stakeholders concentrate on their individual concerns, which are expressed with individual criteria, while the facilitator/analyst focuses on the system overall concerns, which are expressed with system criteria.

There are some differences of decision making activities and paths for the facilitator/analyst and stakeholders, as shown in detail by the interaction names placed beside the lines connecting two classes in the diagrams in Appendix D. Generally, for the decision making of stakeholders, values and concerns are identified and structured by the stakeholders under the guidance of the facilitator/analyst, whose main role in decision analysis is that of facilitation and co-ordination. In the class hierarchy of communication, stakeholders communicate with other decision participants via various kinds of message and get informed about the progress of decision making by the mechanism of decision making progress trace machine. In the hierarchy of problem structuring support, values and concerns of the decision problem are identified by stakeholders under the guidance of the facilitator/analyst. In the hierarchy of problem structuring, a problem is structured by stakeholders. Unlike the facilitator/analyst, who acts mainly as a guide in decision making and deal with the overall problem structure, stakeholders perceive decision problems from individual points of view, which represent different interest parties. Stakeholders identify basic decision alternatives out of decision elements. Individual criteria identified are organised in the form of criterion hierarchies. In the hierarchy of evaluation, decision evaluation at individual levels is carried out by stakeholders. In the evaluation, judgements about the relative importance of individual criteria and the ranking of decision alternatives against every criterion are made. The ranking of decision alternatives can be aided by decision models, which may, for example, calculate a score according to the values of decision attributes for an alternative. The completeness of evaluation data can be examined automatically by a DSS or manually before the aggregation by using achievement measures to obtain individual level evaluation results.

6.3.3 Resources and Paths of Decision making for Domain Experts

Figure 6.3 shows the simplified class interaction diagram for domain experts in the same way as for the facilitator/analyst and for stakeholders. For the detailed class interaction diagram, refer to Appendix D. In fact, Figure 6.3 is identical to Figure 6.2 except for the replacement of the class of stakeholders with the class of domain experts. This is because of two reasons. Firstly, under certain circumstances these two kinds of decision participants may deal with the same kinds (which are classes) of entities during the decision making procedure. This is despite the fact that they have different decision making activities and may interact with different individual objects (may belong to the same classes), which are instances of classes. Secondly, both diagrams are simplified diagrams with omission of class interaction names.

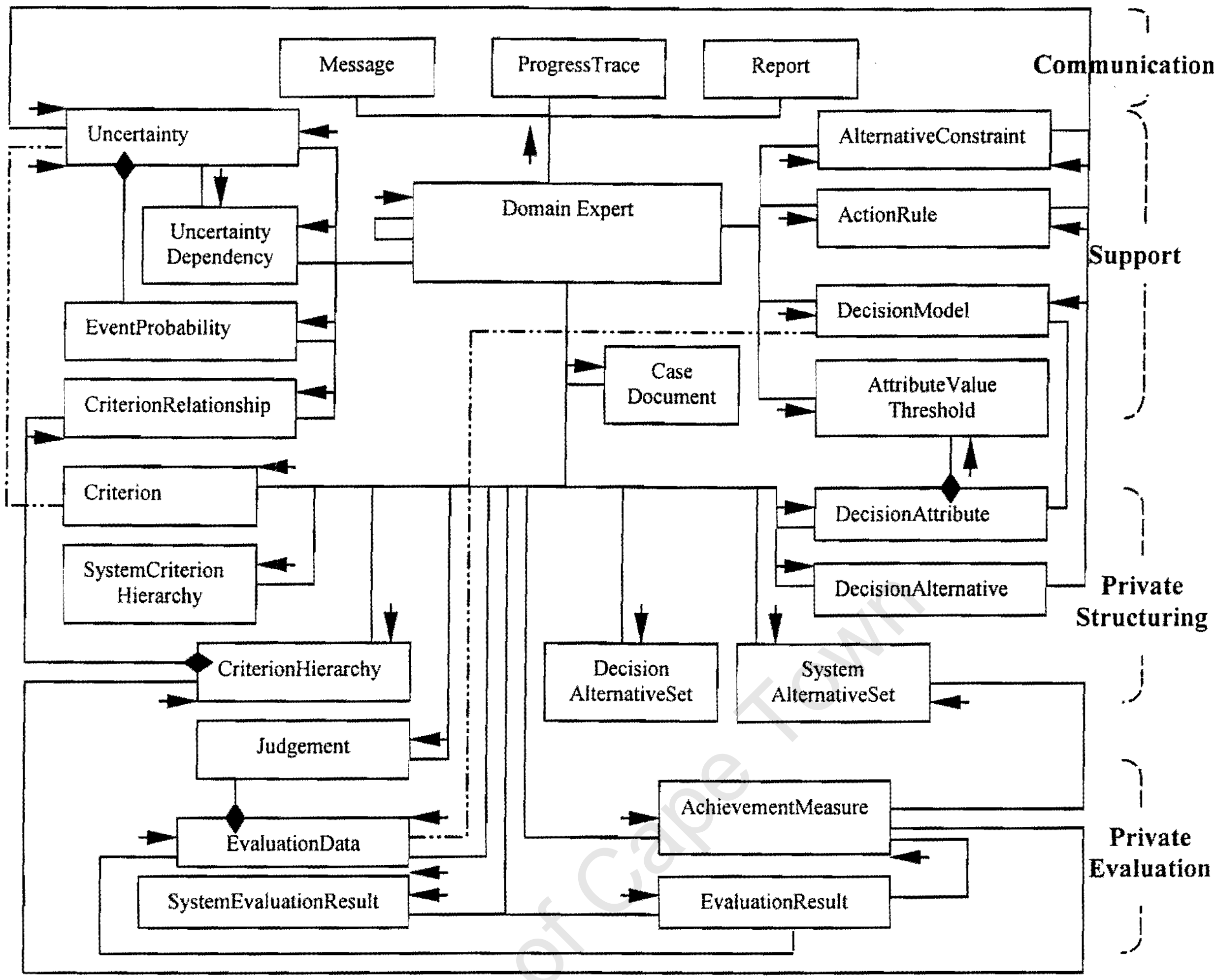


Figure 6.3: Simplified Class Interaction Diagram for Domain Experts

The similarity of the entities included in the class interaction diagrams for the facilitator/analyst, stakeholder, and domain experts discloses some useful insights. Firstly, the facilitator/analyst, stakeholders and domain experts deal with almost the same classes in the decision making procedure despite that they may face different individual objects, which are instances of classes. For example, all of them use criteria in structuring decision problems, but they deal with different instances of the class of criteria. The facilitator/analyst is mainly concerned with system level criteria while stakeholders focus on individual criteria and domain experts may need to consider criteria for their own decision making tasks. Secondly, as discussed in the previous sub-section, with the help of a DSS, stakeholders and domain experts may not need to deal with other decision participants as the facilitator/analysts does during the decision making procedure. Thirdly, every decision participant is encouraged in taking part in the problem identification phase of decision making, which is represented in the class hierarchy of problem structuring support. As discussed in the previous sub-section, it is necessary to take every possible aspect of the decision problem into consideration by all decision participants as indicated by the presence of the same classes in all the class interaction diagrams. Moreover, during the process of problem structuring, stakeholders and domain experts concentrate on their individual criteria while the facilitator/analyst focuses on the system overall criteria. Finally, the facilitator/analyst, stakeholders, and domain experts are all allowed to be able to carry out evaluation at different levels, to have an idea about how the decision result is obtained and to examine relevant values.

Besides their participation in the decision making processes, domain experts may need to carry out their own tasks of decision making to obtain knowledge about some issues that are critical to the decision problem under consideration before they can provide reliable information to other decision participants. Domain experts therefore need to structure their own decision problems and evaluate alternatives and criteria generated for the decision making of these specific issues. This is why the class hierarchies of problem structuring and evaluation called private structuring and private evaluation respectively in the diagram even though not all the classes included in these two hierarchies are for domain experts' own decision making.

The relationships between the four class hierarchies for domain experts are a bit different from those for the facilitator/analyst and stakeholders. The class hierarchies of communication and private problem structuring lay a foundation for problem structuring support and private evaluation respectively. The hierarchy of problem structuring support, nevertheless, does not necessarily help the problem structuring for the domain experts' own decision making although it does for the problem structuring of the system level decision making.

In addition, domain experts mainly focus on providing necessary domain knowledge to other decision participants, while stakeholders are the main players in decision making, identifying various values and concerns and indicating their preferences of decision alternatives under the guidance of the facilitator/analyst. Domain experts may identify values and concerns relevant to the decision problem under the guidance of the facilitator/analyst. They can also check the values and information identified by other decision participants to validate these decision elements for their completeness, accuracy, correctness, and consistency, etc. For example, zoologists may need to check if the survival of certain species of animals that need to be protected is consistent with a decision alternative which contains changed land conditions as a result of land allocation decision. Explorations of uncertainties may be carried out with the help of domain knowledge. Domain experts may also need to examine system level and other individual criterion hierarchies and decision alternatives generated collectively by the decision participants for the overall decision problem. Criterion hierarchies are checked to ensure that all the concerns are taken into consideration. Alternatives may also be checked for some issues as well, such as their feasibility and inclusion of sufficient action elements for the problem under consideration.

6.3.4 Resources and Paths of Decision making for Implementation Agents

Decision implementation may start to be planned after the final decision result is aggregated by the facilitator/analyst and approved by the real decision makers, who are usually governmental organisations. Figure 6.4 is the simplified class interaction diagram for implementation agents. Like the class interaction diagrams for the facilitator/analyst, stakeholders, and domain experts, names of class interactions are omitted from the diagram for the purpose of representation conciseness. For the detailed class interaction diagram, refer to Appendix D.

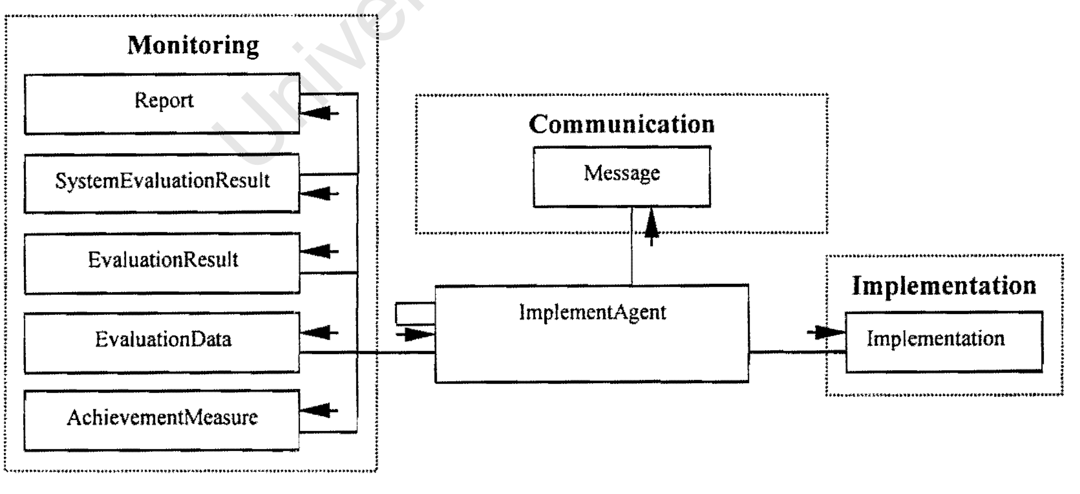


Figure 6.4: Simplified Class Interaction Diagram for Implement Agents

The entities for the decision making of decision implementation agents can be divided into three categories, i.e., communication, decision analysis monitoring and implementation, as shown in the diagram. The communication category mainly includes the class of messages. The category of decision analysis monitoring consists of the classes of reports, evaluation data, achievement measures, evaluation results, and system evaluation results, while evaluation data may contain criterion hierarchies of various levels, alternatives evaluated, judgements made by decision participant at evaluation, etc. The implementation category mainly includes the class of implementation.

Resources of decision making for implementation agents are determined by the entities in the diagram. With the help of a DSS, as for stakeholders and domain experts, implementation agents may only need to communicate with other decision participants virtually by using the system via messages. Data and information about the processes and theories of decision making, as indicated in the classes included in the category of decision analysis monitoring, are needed for examination to understand the analysis results. Implementation of decision results is then carried out with consideration of relevant requirements of resources.

Paths of decision making for implementation agents are determined by the class interactions. Among the three class categories, the most critical one for implementation agents is that of implementation. The category of communication indicates the paths of communications between implementation agents and other decision participants, which may be via messages. Before implementation agents plan decision implementation, they may need systematically to examine relevant decision making data and theoretical grounds for some reasons, such as understanding of the decision result being made and obtaining of confidence about the implementation plan to be made. Various entities are to be checked, including criterion hierarchies of different levels, decision alternatives, judgement and preferences made by decision participants, achievement measures used to aggregate achievements, evaluation results at different times, etc. After the examination, implementation agents can begin plan their decision implementation. Factors influencing the outcome of decision implementation are taken into consideration and an action schedule is constructed. The role of implementation agents is mainly that of planners for the future implementation of the decision.

Although class interaction diagrams give an overall point of view about decision making activities and paths by demonstrating dynamic behaviour of decision analysis, there is a need for further analysis of critical entities of decision problems to identify values and decision opportunities. The next section discusses class state transition, which can further describe the behaviour of the most dynamic classes of decision elements and DSS components.

6.4 Roles of Dynamic Classes for Decision making

There is a need for further exploration of the classes of decision elements and DSS components that possess significant dynamic behaviour in decision making due to the important roles of these classes in the decision making procedure. Firstly, it is important for decision participants to understand what happens inside the entities represented by these dynamic classes. These dynamic classes may undergo dynamic internal behavioural changes over time. A view of internal changes, which are represented by states of dynamic decision classes offers decision participants snapshots of the progress of decision making and may help them carry out further decision making activities effectively. Moreover, the internal dynamic changes of these important classes demonstrate some group decision making paths by the major decision participants for a decision problem since changes inside the classes may be triggered by various events from different decision participants. This is further illustrated with examples in the subsequent paragraphs.

While class operations, which describe static behaviour of classes, and interactions among classes are represented with class interaction diagrams, internal behavioural changes of dynamic decision classes can be modelled by state transition diagrams. The state transition diagram shows the states of a class, the events that cause a transition from one state to another, and the actions that result from a state change. Class state modelling is a valuable means to analyse internal behaviour of dynamic decision entities, specifying the lifetime of these classes. Significant dynamic behaviour of these decision entities may respond to such events as operations, or the passing of time. When an event occurs, some activities will take place, depending on the current state of the class. An activity is an ongoing non-atomic execution within a state. Activities ultimately result in some actions, which represent executable atomic computations that result in a change in state of the class or a return of a message. The state of a class satisfies some conditions, performs some activities, and waits for some events. State transition diagrams model the potential states of classes and the transitions among those states.

State transition diagrams are built for the individual classes along with the analysis of the associated events, states, state transition, and activities. Some events that a state will receive and send are usually already defined in class interaction diagrams as class operations, which are mainly decision making actions. Activities associated with a state are described in state transition diagrams and all the necessary data required is ensured to be available. Each event and the states a class will transit to on receipt of the event have to be identified to construct a state transition diagram in which a sequence of states the class will go through.

By studying class interaction diagrams in Appendix D, classes with significant dynamic behaviour can be identified - those that receive and send many messages, which are resulted from class operations. Five classes are identified as significantly dynamic in the DSS model, which contains the classes of decision elements and DSS components. These dynamic classes are DecsisionAlternative (the class of decision alternatives), DecisionAlternativeSet (the class of individual-level decision alternative sets, which are owned by individual stakeholders and may be used for the generation of a system-level decision alternative set.), SystemAlternativeSet (the class of system-level decision alternative sets, which all stakeholders use for evaluation of alternatives at a final stage). This class is actually a kind of DecisionAlternativeSet but has more specific features), CriterionHierarchy (the class of criterion hierarchies for individual stakeholders), and SystemCriterionHierarchy (the class of system-level criterion hierarchies. It is actually a kind of CriterionHierarchy but has more specific features). However, at the first sight, the classes of facilitators, stakeholders, and domain experts seem to be significant dynamic. It is true that the actors, who are the physical people, have a lot of interactions with DSSs, but the classes modelled in the system do not since these classes carry out few operations in a system. There is a need to distinguish the modelled classes and real actors.

After examining relevant class interaction diagrams in Appendix D, states of these classes are identified. Each of the five dynamic classes has a limited number of states, and can be in only one state at a given time. DecisionAlternativeSet has almost the same states and behaviour as SystemAlternativeSet except that the former does not allow evaluation and ranking, let alone analysis of evaluation results. System-CriterionHierarchy may have the same states but possibly different behaviour as CriterionHierarchy and there is, however, an inheritance relationship between them. Therefore, only three classes, i.e., DecsisionAlternative, DecisionAlternativeSet, and CriterionHierarchy, are needed to be considered for the analysis of dynamic decision making activities.

A graphical representation of states, transition, events, and activities is shown below. A state is rendered as a rectangle with rounded corners. A transition is rendered as a solid directed line. Events are represented as descriptions above the associated line and its conditions are included in square brackets. Activities in a state follow a key word "do/". Self-transitions are represented with a self- pointing line. A transition with no event trigger is called a triggerless transition and there is no need to describe the event since its source state implicitly triggers the transition when it has completed its activities. The initial state is indicated with a filled dot while the final states with a small square. Figure 6.5, 6.6 and 6.7 show the state transition diagrams for Decsision-Alternative, SystemAlternativeSet, and CriterionHierarchy respectively.

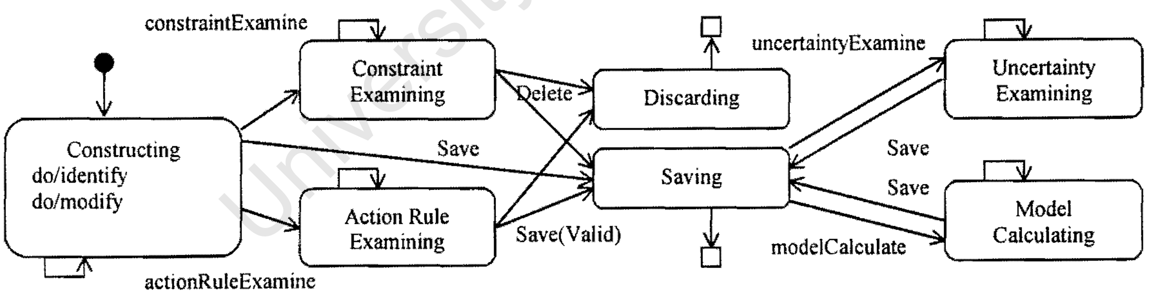


Figure 6.5: State Transition Diagram for DecisionAlternative

Figure 6.5 models the internal dynamic behaviour for the class DecisionAlternative and shows the group decision making paths to deal with decision alternatives. The states that an object (which is an instance of decision alternatives for a specific decision problem) may go through include the following states: constructing from the alternative attributes identified and convening as an alternative, examining with alternative constraints, examining with action rules, saving the alternative, discarding the alternative, calculating with a decision model, and examining uncertainties in the alternative. A decision alternative is first of all identified out of some decision action elements, and it may be modified afterwards. Results from the constructing state of the decision alternative can be saved or be examined with action rules and various constraints for the validity and feasibility of the decision alternative. After the examination, the decision alternative is saved if it is valid or discarded if not. Uncertain areas are then examined to find any decision opportunities, new dimensions, or more considerations for the decision alternative. Modified results are saved. At the same time, decision models may be applied to obtain further information about the decision alternative under consideration. These decision making paths for decision alternatives may be taken by several decision participants.

Figure 6.6 models the internal dynamic behaviour for the class SystemAlternativeSet and shows the group decision making paths to deal with system level alternative sets. The states that a system level alternative set may go through include the following states: creating an alternative set with identified alternatives, cross-checking the set by various participants, modifying, evaluating the set according to a set of criteria, and saving. A system level alternative set is initially created out of identified decision alternatives. It is then checked by various decision participants. Under the state of checking, it may go under validation check to examine if the set contains any invalid alternatives or go under completeness check to see if the set is representative. Results from checking are saved. On the other hand, the initially created alternative set can be modified by relevant decision participants by adding or deleting alternatives in the set. The modified set is saved before it is evaluated according to a criterion hierarchy by a certain stakeholder. It is noticed that some decision making paths for system alternative sets can be taken by different decision participants as a collective effort.

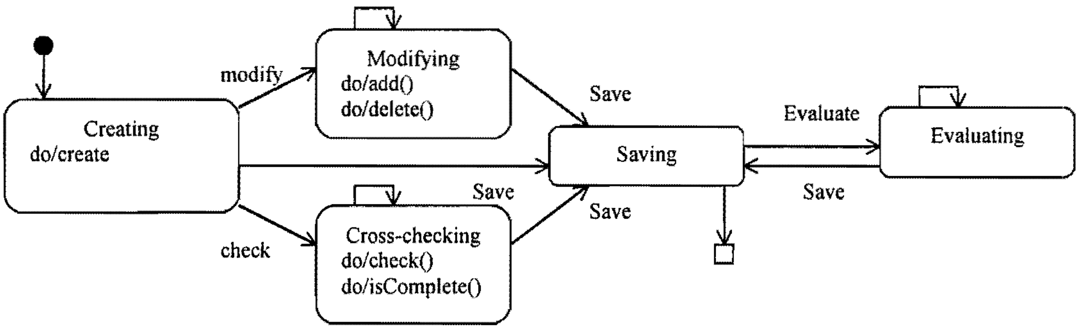


Figure 6.6: State Transition Diagram for SystemAlternativeSet

Figure 6.7 models the internal dynamic behaviour for the class CriterionHierarchy and shows the group decision making paths to deal with criterion hierarchies. The states that a criterion hierarchy may go through include the following states: creating a hierarchy from criteria, cross-checking by various participants, modifying, evaluating, analysing the evaluation result, and saving. A criterion hierarchy is initially created out of identified or unidentified criteria. It is then checked by various decision participants. Under the state of checking, it may go under validation check to examine if the criterion hierarchy is valid in semantics or go under completeness check to see if it contains every possible concerns. Results from checking are saved. On the other hand, an initially created criterion hierarchy can be modified by a certain decision participant. The modified criterion hierarchy is saved together with the newly identified criteria in the hierarchy before it is evaluated by a certain decision participant. The evaluation results can be analysed to check robustness of criterion evaluation. Unlike those for decision alternatives and system alternative sets, the decision making paths for criterion hierarchies are mainly taken by a certain decision participant over time. This, however, constitute the foundation for group decision making since it allows each interest party construct its own criterion structure while decision alternatives are decided as a result of collective efforts.

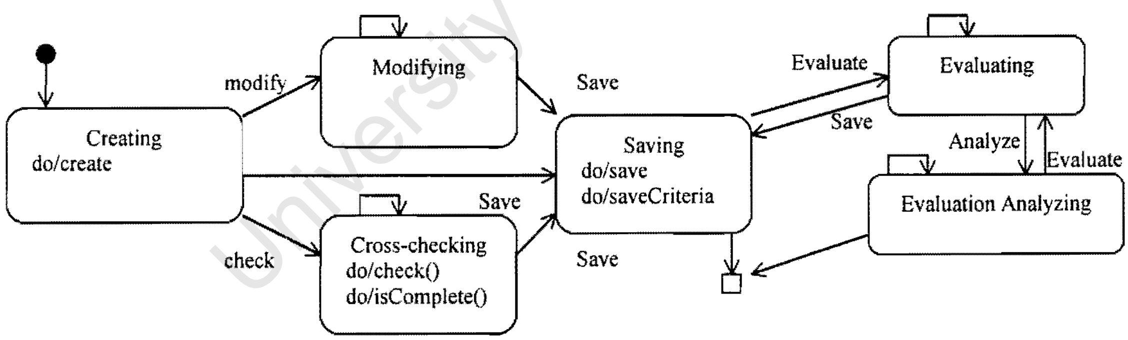


Figure 6.7: State Transition Diagram for CriterionHierarchy

State changes of dynamic classes show the internal transitions of decision entities, while classes model the entities of decision problems and DSSs, and class interactions represent the behaviour of decision analysis resulted from the interactions of these entities. It is noticed that some classes of decision elements and DSS components are closely related and that there are some commonalties among these classes. The next section discusses class relationships, which are an inalienable part of class specifications for the implementation of DSSs and also important for decision analysis.

6.5 Roles of Relationships of Decision Classes

Relationships of decision classes actually provide the conduits for class interactions, which is often referred to class collaboration as one class sends a message to another, as shown in the class interaction diagrams, and also result in behaviours of decision making. There are mainly three kinds of relationships between classes of decision elements and DSS components: general association, generalisations (inheritance) and aggregations. An association is a general semantic connection between two classes, and is represented by using class interactions in the class interaction diagrams. An association name, which is usually an active verb or verb phrase that communicates the meaning of the relationship, are put on the line between the connected (associated) classes in the class interaction diagrams to represent their relationships. They are discussed in the following sub-sections.

6.5.1 Aggregation

An aggregation is an association that shows the relationship between a whole and its parts. Aggregations are usually read as "has a" or "contains". For example, the classes of Facilitator, Stakeholder, and DomainExpert, contain the class of UserConfigure. The relationships between the first three classes and the class UserConfigure are that of aggregation. As shown in Chapter 5, an aggregation is graphically rendered as a line with a filled diamond at the "whole" side of the relationship.

Multiplicity for association and aggregation provides a basic idea of the construction of a decision problem and a DSS, and class interaction in the quantitative aspects. The number of instances that participate in an association or aggregation relationship is referred as the multiplicity for association and aggregation. Two multiplicity indicators might be needed for a relationship. The number of multiplicity is shown on the relationship line at each end. There are mainly four common multiplicity indicators used for the associations (aggregations) of decision classes, i.e., "1" for exactly one, "n" for one or more, "0,1" for zero or one, and "0, n" for zero or more. Problem construction by using multiplicity is demonstrated by examples below. A composite decision class is composed of children classes that contain more specific information. The amount of objects (class instances) may be estimated far before a decision problem or a system is built. This is beneficial to the later stages of analysis and system design due to a better understanding of the problem and the efficient distribution of effort resulted. The multiplicity also helps the actual implementation of class relationships as discussed in the subsequent chapters.

Associations and some aggregations are already explicitly indicated in the interaction diagrams defined. By examining the interactions contained in the use cases of the DSS model, the interaction diagrams, and the classes defined, other principal class relationships are defined. The diagrams shown in this section reveal the aggregations amongst the classes of decision elements and DSS components.

Generally, these aggregation diagrams represent individual decision element construction for decision analysis and a hierarchical or composite point of view to look at some important parts of a decision problem and of a DSS. An aggregation shows the composition for a certain decision class and the aspects to look at when constructing such instances of the class for a decision problem. The following two examples further illustrate the role of class aggregations in decision making.

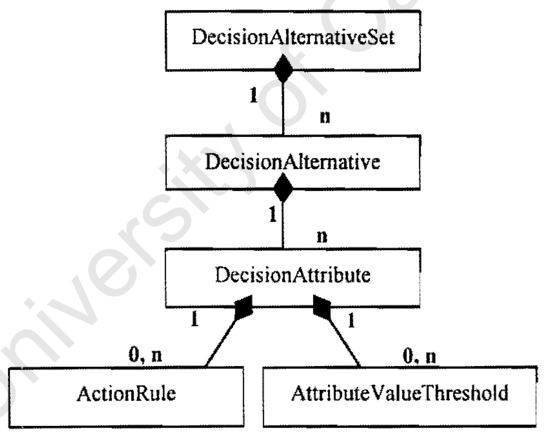


Figure 6.8: Class Aggregations of Decision Alternative Set

Figure 6.8 shows the construction hierarchy for decision alternative sets. As indicated by class relationship multiplicity in the diagram, a decision alternative set is composed of one or several decision alternatives (objects of class Decision-Alternative), while a decision alternative consists of one or several decision attributes (objects of DecisionAttribute). A decision attribute in turn may contain one, several, or no action rules (objects of ActionRule) and one, several, or no attribute value thresholds (objects of AttributeValueThreshold) for that kind of decision attribute. It is obvious that the diagram demonstrates relevant decision elements and the hierarchy to construct a decision alternative set and that the relationships among these entities present a hierarchical point of view for this part of the decision problem, which is the composition of decision alternative sets.

Figure 6.9 basically shows the composition of a decision problem, which is named the class of CurrentCase compared to previous problem cases in the same problem domain, and roughly illustrate the overall point of view of a decision problem. As shown in the diagram, a problem case may contain two different levels of decision entities (such as criterion hierarchies and evaluation results), i.e., system-level and individual interest level, which reflect individual decision making activities and system-level overall aggregation for MCDM in natural resource management. At the system level, there is one or zero instance of a certain decision class for the decision problem under consideration at some stages of decision making, e.g., one or zero system-level criterion hierarchy, evaluation result, and overall alternative set. Zero number of a certain decision entity means that this entity has not been generated at a certain stage. At the individual interest level, there are several or zero instances of an individual-level decision class, each of which is for the decision analysis of an individual decision participant. As shown in the diagram, there are several or zero individual evaluation results, criterion hierarchies, and evaluation evaluations. In addition, a decision problem also includes several or zero decision attributes, which represent the dimensions of decision action to be taken for the natural resource management problem under consideration, while other important entities like decision participants are implicitly contained in the existing entities like individual-level criterion hierarchies.

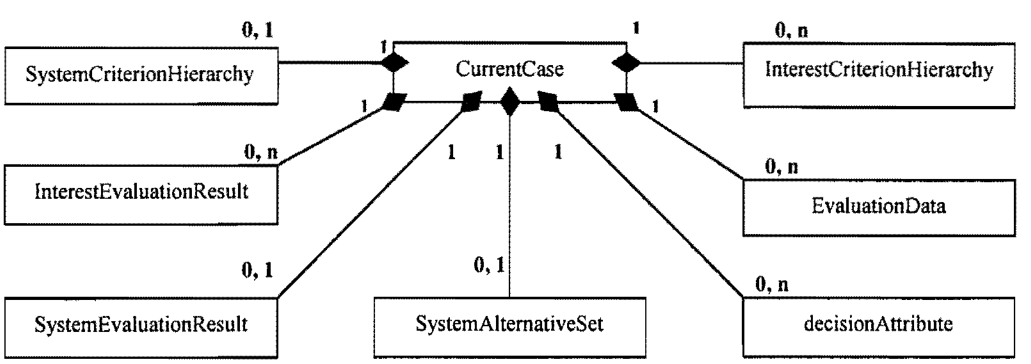


Figure 6.9: Class Aggregations of Current Decision Problem

Other class aggregation relationships are illustrated in Appendix E with diagrams, and they show the composite point of view of different decision entities. As shown in the diagrams, an individual-level criterion hierarchy is constructed by some instances of individual-level criterion with some or without any guidance of criterion relationships, while a system-level criterion hierarchy is by some system-level criterion with some or without any guidance of criterion relationships. A criterion includes one weight and one or no value model. A case document in a DSS is composed of one current problem case and maybe some or no similar problem cases, which has the same composition as a current problem case. A decision participant modelled in a DSS includes a user configuration to configure the functions of a user in the system. An entity of evaluation data may contain one or zero system-level alternative set, criterion hierarchy, and achievement measure as well as some or zero individual-level criterion hierarchies, judgement, and preference. Uncertainty may comprise some or no representations of event probabilities and one or zero uncertainty dependency, which illustrate the factors of the uncertainty. An individual-level evaluation results and a system-level evaluation result may comprise one or zero of final choice of alternative, a ranked alternative list and a sorted alternative set.

6.5.2 Inheritance

Inheritance defines a relationship between classes where one class shares the structure (attributes) and/or behaviour (operations) of one or more classes. It is an "is-a" or "kind-of" relationship. For example, a general class "user" for a computer system may have more specific kinds like "educated user". Inheritance is a relationship between a general thing (called the super or parent class) and a more specific kind of that thing (called the sub- or child class). With an inheritance relationship from the child class to the parent class, the child class will inherit the structure and behaviour of the parent class. The child class may even add new structure and behaviour, or may modify the behaviour of the parent classes. The child class is substitutable for the parent class. Instances of the child class may be used anywhere instances of the parent classes apply. An inheritance relationship is rendered as a directed line with a solid arrow pointing to the parent class.

For the inheritance relationship, generalisation and specialisation are two means often used to identify superclasses and create subclasses. Generalisation means that super classes are created by examining currently existing classes that model the real world and then encapsulating structure and behaviour common to several classes. On the other hand, by means of specialisation subclasses can be created to represent refinements in which structure and behaviour may be added or modified. Subclasses need to specialise the structure and behaviour of an existing class that already exists in an existing internal or a purchased class library. When doing decision analysis and developing a specific DSS, class specialisation is extensively utilised to make use of the classes in the created DSS model.

Therefore, inheritance relationships between decision classes are very important in decision analysis and DSS development. Firstly, inheritance offers a mechanism to reuse existing proven knowledge and past experiences of a similar decision context in decision analysis and DSS development. The real entities of decision elements and DSS components for a specific decision problem and a DSS are generated by the means of specialisation. In addition, inheritance leads to well understanding of the problem and high productivity and high quality of decision analysis and DSS implementation. Generated entities out of well-identified reusable classes are sound in capturing the generic aspects of decision making and flexible in adapting to the new situation of the decision problem under consideration. Finally, inheritance provides a mechanism to manage complexity in decision analysis and DSS development. Structure and behaviour common to several classes are encapsulated and only their refinements are made to specific classes for a certain decision problem or a certain DSS by means of specialisation. This simplifies the representation of the decision problem under consideration with the guidance of existing decision elements and the DSS under construction with the help of the DSS model. The following example of inheritance relationships illustrates the role of inheritance in decision analysis and DSS development.

Figure 6.10 demonstrates the class inheritance relationships between various kinds of decision alternative sets. System alternative sets and interest alternative sets inherit the features of their parent class, decision alternative sets. Generic knowledge of decision alternative sets for decision analysis are kept in these classes. Class DecisionAlternativeSet keeps the most generic knowledge such as the structure and components of a decision alternative set. Class SystemAlternativeSet stores the information about the decision alternative set that can be a system-level alternative set so that the alternatives of its final version can be evaluated by decision participants. Class InterestAlternativeSet represents the relevant information about the decision alternative set that is constructed by a certain decision participant and may need to be examined by other participants to convene a system-level alternative set. These two classes still retain the properties of their parent class (which is Decision-AlternativeSet) but with refinements. Furthermore, as indicated in the diagram, the system-level decision alternative set and individual-level sets for a specific decision problem are generated by means of specialisation of class inheritance. They are child classes of the existing classes with addition of extra details. The proven detailed information stored in the parent classes is encapsulated and may not be concerned if unnecessary. Representation of the decision problem under consideration in this way is simple and easy to be understood since unnecessary details are hidden in the parent classes. Inheritance is therefore able to manage certain degree of complexity in decision analysis and DSS development, and also able to bring understanding of decision problems. In addition, easy adoption to a specific decision problem cases by means of inheritance and easy reuse of generic knowledge lead to effectiveness and efficiency in decision analysis and DSS development.

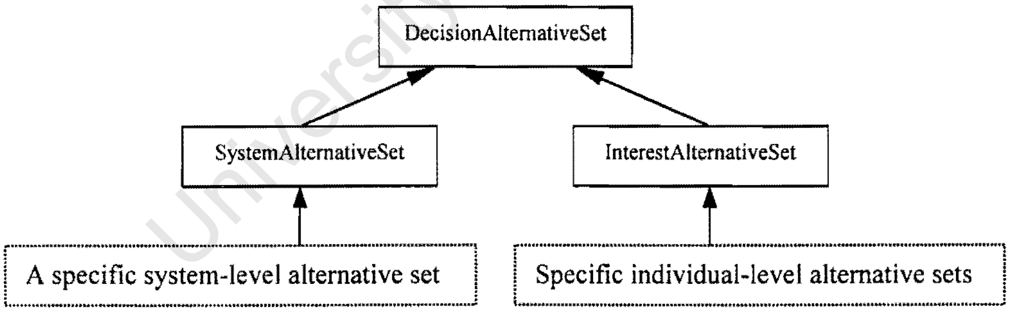


Figure 6.10: Class Inheritance Diagram of Decision Alternative Sets

Appendix F shows the main class inheritance relationships for the DSS model. Users are categorised into two divisions: system administrators and decision participants, while decision participants are further classified into facilitators, stakeholders, domain experts, observers, and implement agents. The classes of system-level criteria and interest-level criteria are child classes of the class of general criteria. Evaluation results include two categories, i.e., system-level evaluation results and interest-level evaluation results. Similarly, criterion hierarchies are divided into two categories, i.e., system-level criterion hierarchies and interest-level criterion hierarchies. Similar problem cases and current problem cases are two kinds of decision problem cases. Preference elicitation is kind of judgement.

Decision classes and their various relationships generically represent decision problems and describe the behaviour of the DSS model. The collective grouping of these classes and relationships constitutes an overall framework, which is described with a system class diagram, for decision analysis and DSS development. This is discussed in the next section.

6.6 System Class Diagram

Classes of decision elements and major DSS components are defined together with their operations, attributes and relationships. They interact with each other to perform decision analysis and the behaviour of a DSS. All the classes are integrated to carry out decision making and DSS functions. A system class diagram provides a useful graphical way to demonstrate the integration and interactions of these classes.

The system class diagram shown in Appendix G is aggregated for decision analysis and the DSS model as an overall class framework, representing overall class structures and class relationships. It offers a mechanism to integrally explore decision making activities and DSS functions resulted from class interactions for decision problems of MCDM in natural resource management since it is simply the combination of previously defined individual interaction diagrams.

When constructing the system class diagram by integrating individual class interaction diagrams, several rules are observed in order to ensure representation consistency. A message is sent from one class to another class only if an association or aggregation connects their respective classes. Two interacting classes in the diagrams have a pathway for communication via either an association or an aggregation. Relationships with sub-classes are expressed via parent classes in the diagram for the purpose of conciseness. Instances of the class DecisionParticipant may interact with each other inside a use case, and reflexive relationships between these instances of the same class are marked. Each class represented in the class diagram participates in at least one use case explicitly or implicitly. Each operation of a class is either used in at least one use case or needed for the purpose of completeness. Each class in the interaction diagrams belongs to the system class diagram unless it represents an external entity. Multiplicity indicators for relationships are showed in most cases except when the relationship is the default case, i.e. "1" to "0, n". This serves to simplify the diagram. To make the diagram easy to read, emphasised and dashed lines are used when lines have to cross over each other.

6.7 Conclusions

Decision elements of decision problems and basic system building components of MCDM DSSs were modelled with classes in object orientation. Classes of decision elements are essentially the fundamental knowledge that can be used and reused for problem analysis and decision making. Classes of decision elements and system building components together with their attributes, operations, relationships and interaction were analysed to illustrate decision making activities in an object-oriented way and also to place a foundation for DSS development.

This chapter has analysed the roles of decision classes and their relationships in decision analysis and DSS development. These classes are very important in many aspects in decision making. Especially, resources and paths of decision making for main decision participants can be determined diagrammatically via class interaction diagrams. The internal state changes of some dynamic decision classes offers decision participants snapshots of the progress of decision making and may demonstrate some group decision making paths by major decision participants for a decision problem. The relationships between classes can be considered in terms of associations and aggregations, which explicitly express the interactions between classes, and of inheritance, which hierarchically construct some classes based the existing definitions of other classes. Class relationships of aggregations represent individual decision element construction for decision analysis and a hierarchical or composite point of view for some important parts of a decision problem and of a DSS. Class relationships of inheritance offer a mechanism to mange complexity and to reuse existing proven knowledge and past experiences of a similar decision context in decision analysis and DSS development, leading to well understanding of the problem and high productivity and high quality of decision analysis and DSS implementation.

The behaviour of decision analysis and DSSs was explored integrally with the grouping of the classes and their relationships. A general framework for DSSs composed of classes of various kinds for MCDM in natural resource management was obtained, in which decision making activities resulted from class interactions were represented collectively. After integrating relevant concepts of object orientation, The next chapter presents methodological guidelines for object-oriented MCDM in natural resource management.

Chapter 7

Object-Oriented MCDM for Natural Resource Management

7.1 Introduction

There are many advantages of object orientation in problem analysis and modelling, as introduced in Chapter 1. Object orientation is able to lead to reusability, extensibility, high productivity, high quality, complexity management, etc, in software implementation as well as in problem analysis.

Our basic thesis is that an object-orientated approach provides two primary contributions to the practice of MCDM:

1) A philosophy to model a decision problem and a DSS in a simple, transparent and natural way. *A decision problem and a DSS are defined by a collection of "objects". The analysis of a decision problem is carried out by organising various "object" around actors. The interactions of "objects" naturally accomplish the system functions of decision making in a DSS.*

2) Reuse of knowledge. *Generic knowledge from both past decision problem cases and the literature is generalised as "classes" (abstractions of the "objects" with common features), which are re-used in a specific case of decision analysis and DSS development.*

These two contributions are described in the subsequent sections in detail. The object-oriented MCDM aims to provide a philosophical methodology for decision analysis, which is able to help understand and articulate values and use them to identify decision opportunities and to create alternatives (Keeney, 1992). It also aims to find an effective and efficient way for decision analysis and DSS development in natural resource management by providing a decision analysis framework and a DSS model.

The objective of this chapter is to describe the methodological guidelines for object-oriented MCDM in the context of natural resource management. There is a strong theoretical foundation of object-oriented decision analysis as object orientation offers a philosophically sound mechanism to model decision problems, decision making activities, and DSSs, as demonstrated in the previous chapters. An overall point of view of the theoretical foundation of object-oriented decision analysis is obtained by integrating those concepts put forward previously.

Based on the theoretical foundation, a general framework of the application of object-oriented decision analysis is proposed to show fundamental ways to utilise object orientation in decision analysis. Methodological guidelines for object-oriented MCDM in natural resource management are presented to provide basic methods with which object orientation can be used in various decision analysis processes, including initial understanding, strategic analysis - brainstorming - decision element identification, structuring and exploring. It is also shown that soft methods of problem analysis from management science may be flexibly applied in the context of object-oriented decision analysis as a supplementary way to facilitate the understanding of a decision problem. DSSs are regarded as an indispensable part of decision analysis nowadays, and can be modelled with the same methodology as that for object-oriented decision analysis. The advantages of the object-oriented philosophy for decision analysis are substantial.

This chapter is organised as follows. Theoretical grounds of object-oriented decision analysis are described in Section 7.2. Section 7.3 outlines the general framework for object-oriented decision analysis while section 7.4 describes in more detail the general practices of object-oriented decision problem structuring of MCDM for natural resource management. Section 7.5 discusses the relationship between decision analysis and DSSs. Object-oriented decision analysis can be implemented with the help of a DSS, which can be developed in an object-oriented way. Advantages of object-oriented philosophy for decision analysis are discussed in Section 7.6. Conclusions are contained in Section 7.7.

7.2 Theoretical Foundation of Object-Oriented Decision Analysis

The theoretical foundation of object-oriented decision analysis derives from its ability to meet the challenges demanded by a philosophically and methodologically sound decision problem structuring method; to carry out decision analysis; and to provide a foundation for the development of DSSs, especially in the cases of natural resource management.

7.2.1 Contribution to Problem Structuring

A method of problem structuring, which is the most critical and most important phase of decision analysis, should be sound in both methodological and philosophical terms, as suggested in Chapter 2 out of the literature review. This kind of method should be able to naturally model the real world in a simple, transparent and flexible way, and should be able to reuse the past experiences and relevant knowledge, being guided by "taxonomies" (von Winterfeldt, 1980). A decision problem can be observed from different viewpoints. In addition, the method should be able to integrate "soft" and hard problem analysis approaches and comprise the four main streams of problem structuring thoughts proposed by (Woolley and Pidd, 1981). It should be able to be applied to a wide range of kinds of problems, especially in those involved with multiple parties dispersed geographically and culturally.

Object orientation has shown its potential to meet these basic requirements in problem structuring and decision analysis. Object orientation offers a philosophy that is able to model decision problems in a simple and transparent way. In modelling the world and the components of it, object orientation assists in organising the functions of various entities around their concepts. Modelling is based on the natural concept whilst postponing attention as to the real detailed functionality. An object view of the world is a natural one, in which the world is viewed as made up of objects and messages. The link between an object and its environment, other objects, is made via messages, which invoke the input and output operations of the object. Objects communicate by passing messages. Objects encapsulate different properties of the entities, such as arguments (attributes) and services (operations or behaviours). The attributes and operations encapsulated in an object can only be accessed by passing messages to that object. Partitioning a decision problem on the basis of objects help manage large complex problems. Besides, details inside an object can be managed at various levels of granularity and can be postponed when necessary. In this sense, object orientation offers a mechanism to manage complexity in decision analysis and DSS development by using classes and class relationships.

Object orientation provides multiple viewpoints to the real world so as to make the problem easy to be understood. Different viewpoints of the problem and a DSS can be observed from various hierarchies of objects. An object is a complete entity which performs a definite task of a decision problem and contains all components needed to carry out the task. A cluster of interacting objects carries out various decision making activities for a decision problem. The decision problem under consideration can be naturally modelled by an object, which in turn is a collection of objects. In addition, different points of view of a decision problem can be observed from decision participants, including stakeholders, a decision facilitator, domain experts, as well as an overall perspective of decision making processes. The provision of multiple viewpoints of a decision problem can help the understanding of the problem.

Object orientation allows the accumulation and reuse of knowledge and experiences in various aspects of decision analysis. In object orientation, objects with the same or similar properties are abstracted as a "class", which is a kind of "taxonomy" (von Winterfeldt, 1980) and is reusable in specific cases of decision analysis. Problems, problem elements and DSS components are classified as classes. An object is an instance of a certain class and would automatically inherit the general characteristics of the class. Reusable classes store knowledge and experiences of decision analysis, and are very important for a specific decision problem to be structured and for a DSS to be developed. Besides, classes may have child classes, which may inherit some features from the parent classes while being able to override other properties. This kind of class inheritance relationship offers a mechanism to reuse existing classes, which keep proven knowledge and past experiences of similar decision contexts, in decision analysis and in DSS development. The real entities of decision elements and DSS components for a specific decision problem and a DSS are generated by the means of class specialisation, which reflect refinements in which structure and behaviour of parent classes may be added or modified. In short, decision elements of decision analysis can be represented in an object-oriented way to construct a basic form of knowledge of decision problem definition, a resource of decision analysis, and a kind of system building material of DSSs. This contributes to the efficient carrying out of the problem analysis and structuring for a specific decision problem.

Object-oriented decision analysis has further advantages as outlined below, which make it desirable for decision analysis.

- Object orientation is able to provide an easy way to bridge the gap between decision analysts, decision makers, domain experts, and DSS researcher for them to understand each other better since object orientation allows all of them to use the same method to explore decision problems and DSSs.
- Object orientation is able to support group decision making in the natural resource management. Multiple interest parties dispersed geographically can be modelled as different "actors" (see Chapter 4), which play an active role in the decision making processes. Their communications and interactions are represented as messages transferred between the objects that represent them.
- Object orientation brings about flexibility to decision analysis in its handling of changes and its ability to adapt to the particular problem on hand via means of reuse and class inheritance. Problem analysis can start from individual objects with a bottom-up approach as well as from an overall point of view with a top-down decomposition approach.
- Soft analysis methods can be applied in an auxiliary manner to strategic problem analysis, which may be based on the existing reusable classes, in identifying objects and specific problem features for the individual decision problem. At the same time, hard analysis approaches can be integrated in individual objects as behaviour of decision analysis.
- Object orientation naturally meets the four main streams of problem structuring thoughts proposed by (Woolley and Pidd, 1981).

As shown in the previous chapters, object orientation can provide a step-by-step procedure for object-oriented problem structuring; an easy way to define decision problems by using objects of decision elements and their relationships; an effective way to understand decision problems through reusable classes; and a people-oriented methodology, which is based on the analysis of various decision participants of a decision problem, for problem analysis. Although many of the above advantages may be realised from other problem structuring methods, object orientation provides a uniform framework for the facilitator to bring many of these together.

Object orientation is able to model both decision making activities and DSS development. The modelling of decision making activities can lead further understanding of decision analysis while a DSS model can be very important to the development of DSSs by improving its effectiveness and efficiency. The following section discusses the modelling of decision making activities.

7.2.2 Object-Oriented Modelling of Decision Analysis and DSSs

The decision analysis of natural resource management problems is perceived as a macro-system, which manages processes and entities to achieve a purpose – to find a satisfactory policy alternative for all the parties involved. Such a system is viewed as an object, which may include a broad range of entities such as decision workshops, communication media, reports, models, analytic tools, and human beings such as facilitators, decision makers, domain experts, other stakeholders, etc. A DSS in the usual sense, which is referred as a micro-system, is another object that in included in the macro-system object to facilitate the decision making procedure. A DSS, either computer based or non- computerised, plays a critical role in the analysis of a decision problem. Data and information from the natural resource considered for allocation and the problem environment and judgements from actors are input to this micro-system object to produce recommended choice and other insights as feedback.

The decision problem environment in a macro-system object is the context in which a decision problem resides. It is also viewed as an object, including a collection of entities that have impact on the allocation of the natural resource under consideration. These entities are classified as interacting categories, each of which is represented as a class. The classification of the entities and their sub-elements of the decision context serves as a guideline for analysis, to ensure that all aspects and issues related to the problem receive the necessary attention in every case. The classes of these entities and their sub-elements can be used as templates for the problem analysis of a specific decision problem.

Decision elements, that is the entities, whether physical or not, which intrinsically exist in a decision problem, are very important in decision analysis. They are an essential part of a decision problem and belong to both the macro-system object and the micro-system object, as the basic elements in the decision making processes and the basis for decision analysis. Each decision element is a potential part of the vocabulary of problem analysis (structuring) and the DSS that will implement part or all of the processes of decision making. A collection of decision element definitions defines the decision problem under consideration. Categories of decision elements are abstracted as classes, which will later help the generation of individual decision element objects for a specific decision problem. The essential classes for decision analysis of a natural decision problem include those of decision attributes, criteria, alternatives, constraints, uncertainties, facilitators, domain experts, stakeholders, decision makers, and other decision participants. Relationships among decision elements are represented as class relationships of aggregation and inheritance. Some composite classes may include components that contain more specific information and some may be classified into several categories.

The general MCDM decision making procedure for natural resource management problems can be modelled with object orientation from different points of view including the interaction of participants and activity sequence, as discussed in Chapter 4. The generic aspects of decision making in natural resource management together with decision elements are represented as reusable classes, which are regarded as an essential part of a DSS that is going to support some or all of the decision making processes. The operations of the classes of decision problems and decision making entities, and the interactions among them model the activities of decision making. The decision making procedure is interpreted as interactive ongoing phases of various activities with different perceptions, representing the views of the different interest groups.

A systematic collection of decision analysis classes, which is critical in carrying out decision analysis with object orientation, is obtained after the comprehensive capture of the functions for DSSs for MCDM natural resource management decision problems (see Chapter 5). DSSs are meant to support all the phases of the decision making procedure. The fundamental functions of a DSS are analysed based on the identification of system actors and the system context. DSS evaluation principles are also applied to ensure that these principles are met by the functions of the DSS under development. General system functions for DSSs are finally represented with use cases and case descriptions. Classes for decision analysis for natural resource management are therefore systematically identified together with their attributes and operations.

Classes and class interactions are able to model the main activities of decision analysis as well as the major functions of a DSS. The link between a class and its environment, other classes, is made via messages. The effect of a message on the class depends both on the message and the receiving class. That is, the same message may cause different actions from two different classes. The action of individual entities and that of the community or of the world as a whole are invoked by the "messages" among the classes. Classes respond to messages by selecting a corresponding operation to execute the message received. The behaviour of a class is a set of operations and mainly represents various decision making actions. The interaction of classes carries out certain decision making activities. For a specific decision problem, the decision making activities are carried out by specific objects, which are instances of the classes that generically represent objects of the same feature.

The analysis of the objects that manifest significantly dynamic behaviour in decision making offers decision participants snapshots of the progress of decision making and may demonstrate some group decision making paths by major decision participants for a decision problem, as discussed in Chapter 6. It is important for decision participants to understand what causes the internal state changes of some important decision entities during various phases of decision making as it may help them carry out further decision making activities effectively by co-ordinating various decision making activities from different decision participants.

The analysis of class relationships, which is discussed in Chapter 6 in detail, is also useful in decision analysis and DSS development. The class relationship of aggregation is able to offer a mechanism to construct specific decision entities for a decision problem, and to provide a hierarchical or composite point of view for some important parts of a decision problem and of a DSS. An aggregation shows the composition for a certain decision class and the aspects to look at when constructing such instances of the class for a decision problem. At the same time, the class relationship of inheritance provides a mechanism to manage complexity and reuse existing proven knowledge and past experiences, leading to good understanding of the problem and high productivity and high quality of decision analysis and DSS implementation.

A general framework for decision analysis and the modelling of DSSs for MCDM in natural resource management is then obtained by integrating various classes and class relationships into a single diagram. It offers an integral object-oriented way to study decision analysis and DSSs. The next section further discusses the application of object-oriented philosophy in decision analysis by presenting a general decision analysis framework.

7.3 General Framework for Implementing Object-Oriented Decision Analysis

A general framework of object-oriented decision analysis is shown in Figure 7.1. This figure demonstrates how the general process for the methodology of object-oriented decision analysis, especially of problem analysis and structuring, can be applied by facilitators and DSS system developers. Figure 7.1 is very similar to Figure 3.1, which demonstrates the general diagram for object-oriented problem analysis and structuring. These two figures are based on the same fundamental idea, which is that *object orientation offers a philosophy for problem modelling and a model of decision problem definitions for a specific decision problem to be defined.* The constructed model is instantiated – basically by specialising all the classes in the model to specific objects. This instantiation offers an initial definition to the decision problem under consideration and may help understand the problem, getting decision participants to have a basic knowledge of the decision making issues. Various decision making activities are accomplished by operations of objects and the interactions between them, which can be easily implemented in DSSs. However, Figure 7.1 takes the full support of decision analysis by object orientation into consideration. It depicts decision making processes of object-oriented decision analysis explicitly.

The decision making process model included in the object-oriented decision analysis recognises four main phases, i.e., initial understanding (IU), strategic analysis - brainstorming - decision element identification (SBI), structuring, and exploring, as shown in Figure 7.1.

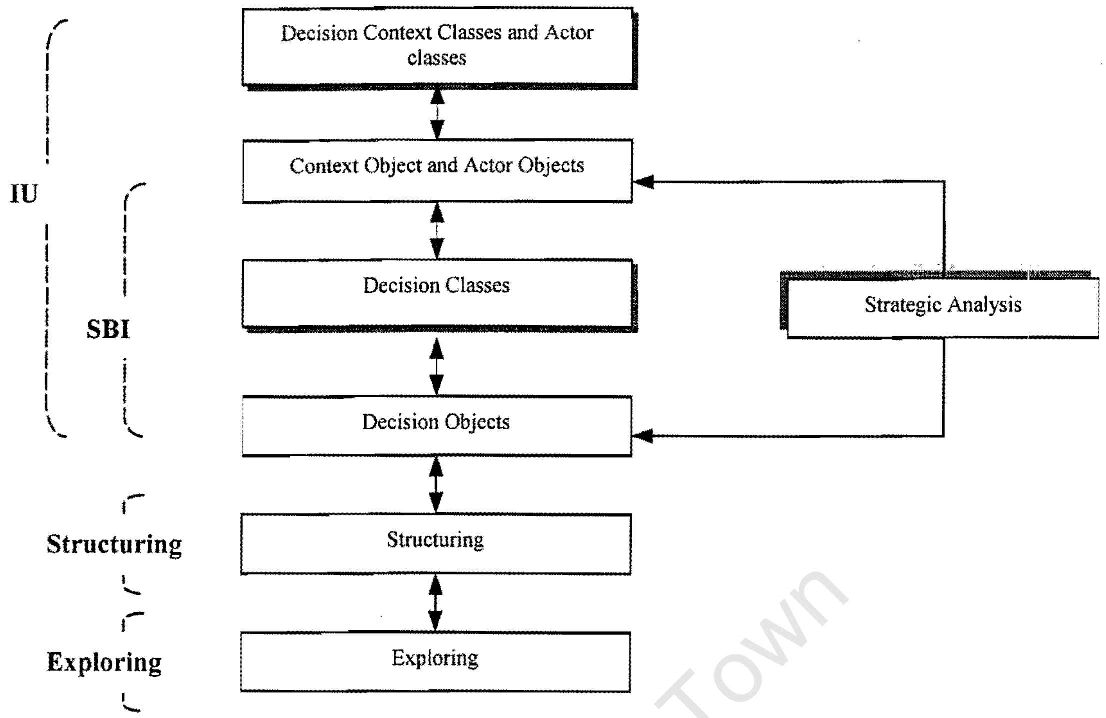


Figure 7.1: Object-Oriented Decision Analysis

In the phase of initial understanding, the decision context and actors, who have an active role in decision making, are identified by means of instantiation of relevant classes. Classes are generated beforehand out of relevant knowledge and past experiences for the problem domain, which contains the problems with similar feature to the problem under consideration. At the same time, other decision classes are also instantiated as specific objects for the specific decision problem. Initial understanding of the decision problem concerned is achieved as a result of the analysis of the decision context, the actors involved, and other decision entities.

The phase of strategic analysis - brainstorming - decision element identification (SBI) seeks to understand the decision problem under consideration. It may be based on the initial identification of decision analysis objects, and work on the same set of decision entities as that for the process of initial understanding, as indicated in the figure. Other techniques of strategic analysis and brainstorming can also be utilised as supplementary approaches to obtain a thorough understanding of the decision problem under consideration and to produce a complete and refined collection of decision elements for the structuring of the problem. This phase can also be carried out without any existing classes of decision analysis at the very first stage. This allows object orientation to be able to be applied in cases where no previous similar decision problems have been analysed.

Structuring of a decision problem is then started after the identification of various decision elements with the guidance of problem structuring knowledge stored in relevant classes. Hierarchies of decision criteria are created out of the structures offered by classes of decision criterion hierarchies. Decision alternatives are generated out of the domain knowledge kept in various classes such as action elements and constraints.

Exploring of a decision problem starts after the structuring of the decision problem. It explores preferences and judgements of decision participants, and studies various outcomes of selection by values or by actual impact on interest parties involved in the decision problem, which may be obtained by implementation planning, or even partial implementation, of the decision made. Feedback may be obtained and further exploration of the decision problem may be needed. The phase of exploring in object-oriented decision analysis actually contains the processes of evaluation, choice and implementation included in some decision making process models, which are reviewed in Chapter 2. Actually, all decision making activities contained in the phase of exploring are carried out by the operations of various classes of decision analysis and the interactions of these classes, which can be easily implemented in a computerised DSS. Therefore, exploring of decision problems in object-oriented decision analysis usually involves the development and the assistance of a DSS.

The contribution of an object-oriented MCDM approach to decision analysis is that "objects" of a decision problem are naturally modelled in a simple and transparent way and then used through all phases of the analysis. Generic knowledge from past decision problem cases and the literature can be reused in decision analysis by instantiating various "Classes".

It is noted that the decision making process model of object-oriented decision analysis resembles to some extent that proposed for the purpose of DSS study by Finlay (1994). Finlay's model is claimed in turn to be a "slightly modified and extended" version based on Simon's (1965, 1977) proposal of intelligence, design, choice, and review. It consists of three phases, i.e., structuring, understanding and action. The structuring phase detects and defines the decision problem concerned. The understanding phase includes detailed DSS design, exploring course of action, and decision taking. The action phase is composed of decision implementation and review. Finlay (1994) placed much attention on the role of DSSs in decision analysis as a factor to bring understanding to decision problems. However, there may be some arguments (see Chapter 2) about the sequence of the phases of structuring and understanding and also about the contribution of DSSs to the problem structuring.

On the other hand, the decision making process model of object-oriented decision analysis regards DSSs as a potential all-round player in decision analysis, being able to contributing to all the phases of understanding, which includes initial understanding (IU) and strategic analysis - brainstorming - decision element identification (SBI), structuring and exploring. However, DSSs are able to contribute more greatly to the phase of exploring than the other two since decision activities of exploring can be modelled with operations and interactions of classes, which can be easily implemented in a computerised DSS. Besides, in object-oriented decision analysis, understanding is regarded as the basis of problem structuring.

Actually, the understanding of a decision problem never stops. The phase of exploring may reveal new findings concerned with the problem. Decision participants constantly iterate amongst the processes of decision making, making revisions and bringing to their attention possible conflicts and inconsistencies as new insights are obtained and more knowledge about the problem is gathered in each interaction. The next section discusses the practices of these processes applied in practical MCDM decision problems for natural resource management.

7.4 The Practice of Object-Oriented Decision Analysis

Systematic application of object orientation in decision analysis for MCDM in natural resource management may be carried out under the guidance of the general framework outlined above. Concepts of object orientation, such as classes and class interactions, are applied in the four processes of decision making, including initial understanding (IU), strategic analysis - brainstorming - decision element identification (SBI), structuring and exploring. We now illustrate the implementation of this framework by reference to the decision problem of land use for exotic forest plantations (Stewart and Joubert, 1999), a simplified form of which was used in Chapter 5 to demonstrate DSS functions.

7.4.1 Initial Understanding (IU)

The process of initial understanding, which is regarded as the first sub-process of problem understanding, is a search process involving searching existing generic aspects, such as decision contexts and decision participants, of decision making in the problem domain of the decision problem under consideration for various information and knowledge. This may be based on the idea of reusable classes of object orientation. Classes represent existing knowledge and experiences of decision analysis while objects are related to the specific problem situations. Generic aspects of decision making are various classes stored on files and in computerised DSSs. A full list of classes is included in Appendix B. These classes provide generic information and knowledge to the analysis of the decision problem concerned.

There are two basic ways to reuse the classes of decision analysis. The first one is via direct instantiation of classes. Instances of a class automatically inherit the characteristics, such as attributes and operations, of the class. The instances of a class are objects, which share the same set of attributes and operations, but with different values and status of operations. For example, a forestry company and a water affairs department can be regarded as two instances of the class of "stakeholder", inheriting the attributes and operations from the class but possessing their own individual values. Direct instantiation usually happens when a class can represent the properties of individual entities in some specific details. The other way of class reuse is through specialisation and instantiation. Specialisation allows the generation of child classes out of existing classes, which abstract the generic features of the same decision entities. Child classes may inherit some properties from the parent classes while being able to override others, and may also have some special properties of their own. The child classes generated can represent specific characteristics of certain decision entities for the decision problem concerned. Specific decision entities for the decision problem are then created by means of instantiation of the child classes. For example, the criterion classes of "timber industry" and "tourism industry" may be considered two child classes generated from a parent class of "economy", inheriting some features of the parent class while adding some detailed properties of their own. The two child criterion classes may then be instantiated as specific criteria of "timber industry" and "tourism industry" for a decision problem.

Decision context classes and actor classes can thus make important contributions to initial understanding. A decision context sets the boundaries of a decision problem along with its interfaces to the external world, including influence factors and other decision environment elements. A decision context can explain the underlying dynamics of the decision problem as it is the fundamental environment in which the decision problem resides and it shows all the possible impacts of and influences on the decision problem. Actors are physical human beings who have influence on or be influenced by the decision problem under consideration, including those who have active roles to play in decision analysis (these actors are called decision participants). Actors are considered critical and the most important elements in object-oriented decision analysis since decision problems are defined according to their individual perceptions of the real world, which are based on their understanding of the problem. The identification of the decision context and the actors for a decision problem involved can greatly contribute to the understanding of the problem and can lead to the further identification of decision elements.

Objects of decision contexts and actors for a decision problem are initially obtained from their classes, which can be created out of the literature and/or past experiences. The decision context for a specific problem is an instance of a corresponding decision context class for decision problems, which are of similar features to the problem under consideration. The actors involved in the decision problem are the instances of the actors classes generalised for similar decision problems. The derived decision context provides basic knowledge for the decision environment and the properties of the derived actors give a clue about where to find more information. In short, the decision context and the identified actors are the basis of further decision analysis.

As an example of class reuse, the class of actors is discussed in detail here to show the usefulness of classes. There are different ways of actor classifications in decision analysis. Basically, actors can be a physical person or group of persons who hold the same stake, such as decision makers, facilitators, domain experts, stakeholders, organisations, and other related parties. Stakeholders are very important actors, who either affect or are affected by the natural resource concerned in some way. Some of these actors, such as some domain experts and stakeholders, are decision participants, who play an active role in decision making processes. As a result of the study of real practical cases (Lootsma, Meisner and Schellemans, 1986; Stewart, 1988; Stewart and Brent, 1988; Mendoza, 1988; Glover and Martinson, 1987; Hallefjord and Jornsten, 1986; Sandiford, 1986; Stewart, Joubert, Scott and Low, 1996; etc), actor classes for natural resource management decision problems may be seen to include domain experts, governmental organisations, non-governmental organisations, local communities, neighbour communities, and decision facilitators, as shown in Table 7.1. Domain experts are the experts offering domain knowledge in various aspects. Governmental organisations hold political policies and make a final physical decision to implement the choice of the decision analysis. Non-governmental organisations have the same interest related to the decision to be made. Local communities may affect or be affected directly or indirectly by the decision of natural resource management. Neighbour communities may also affect or be affected directly or indirectly by the decision but are geographically apart from the site where the natural resource concerned is located. Decision facilitators play a role to co-ordinate the whole procedure of the decision making. Governmental organisations, local communities, and neighbour communities are collectively called stakeholders in natural resource management decision problems.

Table 7.1: Actor Classes for Natural Resource Management Decision Problems

Actor Class Name	Brief Description
Domain experts	Offering domain knowledge
Governmental organisations	Holding policies and making a real decision
Non-governmental organisations	Social groups having the same interest
Local communities	Local communities related to the decision
Neighbour communities	Communities geographically apart from the site
Decision facilitators	Co-ordination of the decision making procedure

These classes of actors are defined with their generic attributes and behaviours in decision analysis. For a specific decision problem, individual actors can be preliminarily obtained through these classes. In addition, the relationships among these individual actors can also be roughly understood under the guidance of the existing classes of actor-playing diagrams, which depict various generic interactions among actors. Classification of actors obviously brings about initial understanding of a decision problem.

Further understanding of a decision problem during the process of initial understanding is achieved by studying other decision classes. These classes are mainly those of decision elements even though there are some DSS-oriented classes, such as files and data, for the development of DSSs. Decision elements include the entities of the natural resource concerned, the entities that take part in the processes of decision making, and those that impact and/or are impacted by the decision made. Various concerns (decision criteria), decision alternative, and decision participants are examples of decision elements. A collection of decision elements is basically the definitions of decision problems. Decision element classes represent categories of decision elements with similar attributes.

To obtain generic information and knowledge about a decision problem, it is very helpful to examine these classes as they keep relevant information and knowledge about the problem domain, also obtained from the literature and/or past experiences such as the classes of decision contexts and actors. *Decision entities for a specific decision problem can be initially obtained via means of instantiation and specialisation of the decision classes. The instances of decision classes (or child classes) are decision objects, which represent individual decision entities for a specific problem, which define various aspects of the decision problem.* Preliminary definition and further initial understanding of a decision problem is gained as a result of the examination of these decision entities.

Example:

In the decision problem of land use for exotic forest plantations (Stewart and Joubert, 1999), whose simplified form was used in Chapter 5, the decision context for the decision problem is first identified as a kind of land use for plantations of exotic trees in a particular district. This kind of decision context contains various interactive factors, including economic value, natural ecosystems, water run-off patterns, and river flows.

Actors are instantiated from the classes for natural resource management decision problems, such as domain experts, governmental organisations, local communities, neighbour communities, and decision facilitators. In the specific context discussed by Stewart and Joubert (1999), domain experts were found to include the Department of Nature Conservation of the University of Stellenbosch and the Division of Environmental Technology of the South African Council for Scientific and Industrial Research (CSIR). The initial stakeholder appeared to be North East Cape Forests (NECF), which is a forestry company. The actors of organisations and other communities apart from the forest company were instantiated under the guidance of their respective classes. The actors of governmental organisations included the national Department of Water Affairs and Forestry, the provincial Department of Nature Conservation, and the provincial Department of Agriculture and Land Use Planning. The actors of other local communities included the town council of the Maclear district and the local farmers.

Other decision entities are also identified such as influence factors, decision action elements, uncertainties, criteria, etc. Under the guidance of existing influence factor classes, primary influence factors were identified. These were population pressures in the social class, high unemployment and poor neighbour district in the economic class, rapid upliftment of the previously severely disadvantaged black communities by the government in the political class, and one of the three most threatened habitats for plant species in Africa (Afforestation, overgrazing and increased crop-farming), run-off and prolonged drought in the environmental class. Two important decision action elements were also identified, namely the proportion of afforestation of the area expressed as percentage and the types of trees to be planted, i.e. pine, bluegum, or mixed. Some criteria were initially identified by examining existing criterion classes and influence factors. Examples were wild life conservation, employment, water supply, stability of income, stability of industry, etc. Examples of basic uncertainties included those of employment, impacts on secondary industries, and conservation extent.

After the initial study of the decision context, actors and relevant decision entities, a preliminary understanding of the decision problem could be obtained as described above.

The emphasis in this phase is learning by studying existing "classes" that represent generic knowledge from experiences and the literature, and reusing them to identify initially individual "objects" which are the philosophical base of object-oriented MCDM to model a decision problem.

A rough understanding about a decision problem will be obtained during the initial understanding process. Instantiation and specialisation of existing classes are by no means complete and correct. Much of this work has to be left until a comprehensive understanding is achieved. The next section discusses the second sub-process of problem understanding.

7.4.2 Strategic Analysis - Brainstorming - Decision Element Identification (SBI)

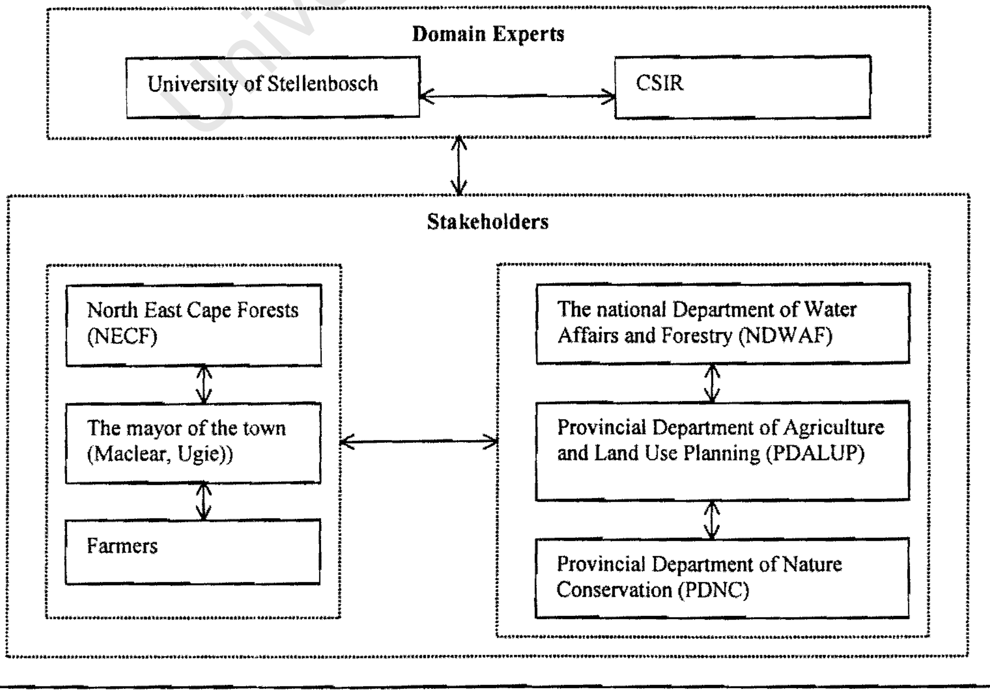
A comprehensive understanding of a decision problem is sought during the process of strategic analysis - brainstorming - decision element identification (SBI). Strategic analyses are carried out based on the initial understanding phase to bring new insights to a specific decision problem. Detailed information and problem specific values are found out. Elements of action, different level of concerns, and aspects of uncertainties are perceived by various decision participants. They are elicited through brainstorming techniques later on to help the identification of extra decision elements and to complete the definitions of previously generated decision entities before the actual structuring of a decision problem. The phase of strategic analysis - brainstorming - decision element identification (SBI) is a process of study and observation before the actions of structuring and exploring.

Once again, the object-oriented approach can be used to facilitate this phase of the analysis, which is a strongly human-oriented process. The roles of actors created in the previous process as instantiated instances of existing actor classes need to receive particular attention in the decision context derived previously from its class. The decision context shows the boundaries of a problem and interactions between different entities. Various impacts of the decision context on actors, and influences that society may have on the decision context, are examined in order to identify the possibilities for new actors. These are then usually discovered in conversations with existing actors such as stakeholders and domain experts. Problem analysis thereafter is based on the strategic analysis of identified actors. According to Checkland and Scholes (1990), "purposeful action" to impact the decision context is a feature role by human beings in decision problems, whether in the public or in the private sector, whether in small firms or in giant organisations. The roles played by human beings are critical in decision analysis.

Analysis of actors starts with their relationships. Actors are defined based on the initial instances generated previously under the guidance of their respective classes. After the identification of major actors, an actor-playing diagram for a decision problem can be generated via means of instantiation and specialisation of the corresponding actor-playing class.

Example: The preliminary actor-playing diagram for the land use case

In order to show their preliminary interactions, the actors identified in the previous process are diagrammed under the guidance of the generic actor-playing diagram, which is demonstrated by Figure 4.5 in Chapter 4.



An actor-playing diagram, which shows the interaction of actors, is able to help analysts understand preliminarily the relationships among actors, find extra actors, and concentrate on the key actors in the analysis of a specific decision problem.

Strategic analysis of actors is based on the available understanding of a decision problem, which can be enhanced through class reuse, and is then mainly focused on the roles played by individual actors. Such strategic analysis can be carried out in an object-oriented way. The classes of decision elements, the component elements of decision contexts, and actors serve as a guideline (see Section 7.2.2 and Chapter 4 for detailed discussion) for analysts to ensure that all aspects and issues related to the problem receive the necessary attention in a case. They are used as templates for the decision analysis of a specific decision problem, bringing initial understanding and preliminary object definitions. Individual decision entities, which intrinsically exist in a decision problem, are treated as objects, which have their own behaviours and attributes and can interact each other to accomplish decision making tasks. The aim of object-oriented strategic analysis is to define comprehensively individual objects for a decision problem with the assistance of further understanding of the problem resulted from close studies of existing roughly defined objects and their interactions. The analysis is carried out in an incremental way. That is to say, a later version of analysis is a refinement of a proceeding step. Examinations of objects are constantly iterated, making revisions as more information about the problem is gathered in each interaction. This finally results in a definition of a decision problem by a collection of objects of various entities, including decision elements and those elements included in the decision context.

In Chapter 4, the component elements of a problem context for natural resource management decision problems were classified as four interacting categories: influence factors, uncertainties, resources, and rules, which are represented as four classes in object orientation. Resources contain various kinds of physical materials, such as equipment, land and water, and non-physical issues, such as technology, finance, authorities, time and human, which are consumed or produced as a result of decisions made and by the consumption of the related natural resources. Uncertainties refer to the uncertain aspects of a problem environment. Rules are the regulatory governmental acts and contracts between agencies that may control and influence the use and distribution of various resources. Influence factors are various elements that have influences on or are influenced by decision making activities. These decision context components are the basic sources of perceptions of actors towards their subjective view of a decision problem since actors perceive the real world through interactions with its environment.

Influence factors need to be discovered in conversations with actors such as domain experts and stakeholders, but in conjunction with examination of the existing classes from previous analyses in similar problems. By examination of previous examples (Chankong and Haimes, 1985; van Pelt, 1993; Perrings, 1994; Faucheux and Froger, 1995; Stewart, Joubert, Scott and Low, 1996; etc.), influence factors may generally be classified into four categories: social, economic, political, and environmental factors, each of which may be represented as a class. For each class of the influence factors, there are further subclasses of factors to be considered. Primary influence factors may be identified after consultations with domain experts. More factors will emerge later during the strategic analysis. An incremental procedure is usually adopted in the identification of influence factors as new factors turn up in the process of analysis.

It is important to explore all areas of uncertainty in a decision problem. Three kinds of uncertainty are observed in natural resource management decision problems, as discussed in Chapter 4. The first kind of uncertainty is the imprecision of human minds and their expression about subjective judgements, with no formal probability theory for its solution. The second kind of uncertainty is about the unreliability of data due to lack of sufficient information. The third kind of uncertainty is resulted from unknowable random processes with possible statistical distribution theory. Basic knowledge of uncertainties is kept in classes, which may be used during the strategic analysis to find and analyse uncertain areas. A working list of uncertainty areas is progressively built up from the early stage of analysis. Uncertain areas are identified by examining suggested elements of a decision problem for any uncertain aspects in them. Possible outcomes of a certain uncertain area are represented as uncertain scenarios so as to keep options open for later resolution. The analysis of uncertainty areas identified from the initial analysis proceeds when the strategic analysis carries on. With the assistance of more information and additional analysis; some scenarios become clear. At the same time, various approaches of uncertainty analysis are applied to deal with the management and resolution of individual uncertainty areas.

It may be easier to identify rules and resources for most decision problems than influence factors and uncertainty areas. Rules can be found in the documents of governmental acts and organisational files concerning the use and distribution of the natural resources under consideration. Several classes of rules provide a guide for the identification of individual rules for a specific decision problem. These classes include those of laws, contracts and other legitimate documents. Resources in a decision problem environment are mainly referred to the materials of consumption and production during decision analysis. There are general classes, i.e. physical materials and non-physical materials, each of which in turn contains some sub-classes. The class of physical materials includes subclasses of equipment, physical materials, and natural resources while the class of non-physical materials consists of sub-classes of technology, finance, authorities, time, and human labour. Resources are identified by considering what are needed for decision making process and for the usage of the concerned resources, and what are the products of such usage.

The other important decision entities are decision elements. It is noted that some component elements of decision contexts may be turned into decision elements at a later stage of decision analysis. Actually, the analysis of the decision context for a specific decision problem brings about the identification of some decision elements.

Decision elements are the entities, whether physical or not, that intrinsically exist in a decision problem and are related to decision making processes. Examples are stakeholders, criteria, criterion hierarchies, and decision alternatives. Decision elements are the basis for further decision analysis, and the collection of their definitions defines the decision problem under consideration. A proposed set of classes for decision analysis of a natural resource management decision problem has been generated, and is included in Appendix B. Their roles in decision analysis are discussed in detail in Chapter 6. Decision element classes represent abstracted categories of decision elements with same features while decision element objects, which are created via means of class specialisation and instantiation, represent individual decision elements for a specific decision problem. The perceptions of various decision elements by decision participants are thus initially achieved. Brainstorming at a later stage will obtain a comprehensive collection of defined decision elements.

Identified objects of decision entities, including influence factors, uncertainties, rules, resources, decision elements, and actors, are then organised and put in the analysis context of each actor. The actor interacts with various decision entities, playing an active role in decision analysis. Figure 7.2 shows the general idea of actor-oriented analysis. Various decision entities are considered for a certain actor for their interactions (indicated with solid arrow-headed lines) with the actor. At the same time, these entities may have interactions (indicated with dotted lines) among themselves to meet the behavioural requirements of the actor. Behavioural analysis of actors organises relevant decision objects together to provide an overall point of view of a decision problem to the actors and also to offer a systematic mechanism to carry out various decision making activities of the actors. Definitions of decision objects are refined and/or completed by examining their operations and attributes, through which decision making activities are carried out. Actor-oriented strategic analysis can bring further understanding to the decision problem under consideration, leading to clear definitions of the decision objects for the decision problem.

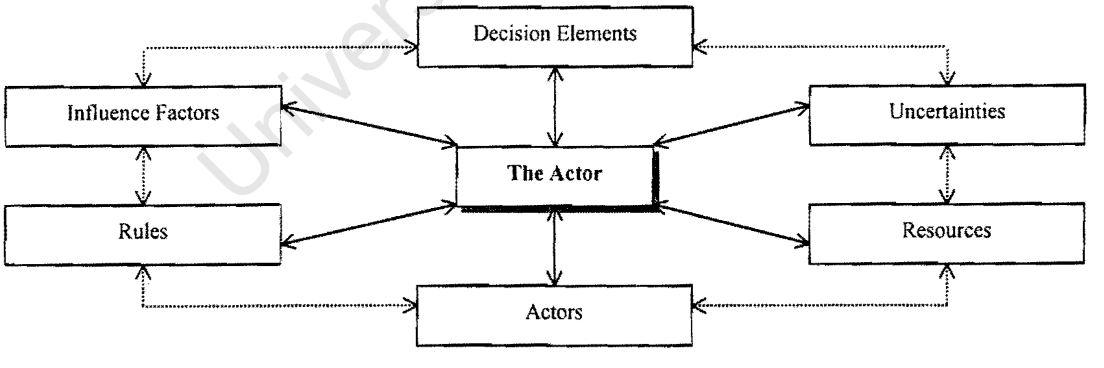
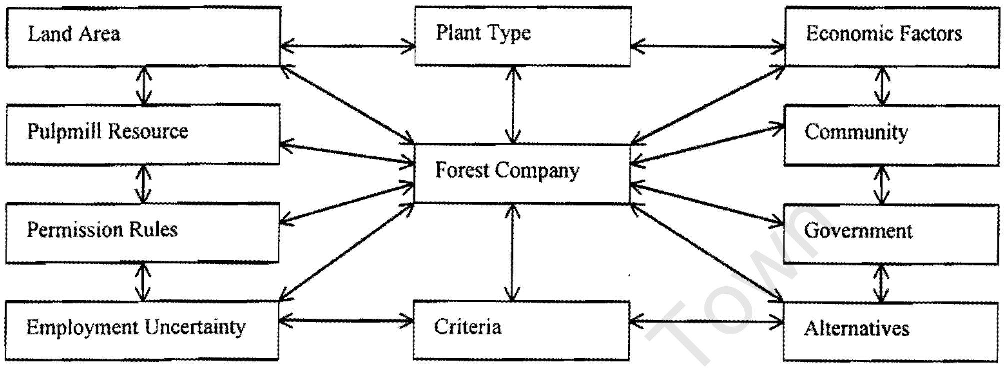


Figure 7.2: Actor-Oriented Analysis

Different viewpoints from various decision participants, including stakeholders, decision facilitators, domain experts, are observed as well as an overall perspective of decision making processes. The group structure of decision participants and the group interaction give the overall viewpoint of the decision problem. A group of decision participants communicates with each other under the guidance of an analyst/facilitator, who is skilled in the processes of group discussion and decision analysis. The analyst/facilitator also assists decision participants in brainstorming, structuring, and exploring the decision problem later.

Example:

A simplified strategic analysis based on object orientation is illustrated for the forestry company actor.



The forest company needs to acquire an amount of land in the region by seeking government permission for plantation of some kinds of commercial plants. The afforestation needs to be economically sustainable in order to produce timber to support the operation of a pulpmill. The concerns of the company are ultimately expressed with the definitions of various criteria on which decisions regarding the desirability of decision alternatives could be based.

This methodology of strategic analysis can thus usefully be based on the concepts of objects and their interactions. Individual objects for a decision problem are identified and defined, and their interactions are then analysed to obtain an overall point of view about the problem. These objects may be initially created by means of instantiation and specialisation from their corresponding classes. Templates of class interactions are also recorded along with the definitions of classes. Classes and their interaction templates are able to keep relevant knowledge and experiences from the literature and previous cases. They offer knowledge reuse, one of the major benefits from object-orientation.

The existence of pre-generated classes and interaction templates is not a requisite for the object-oriented strategic analysis. The methodology can be used in the very first problem case, in which there are no existing classes. In this case, all objects for a decision problem have to be created from scratch, which is a strenuous and error-prone process due to the number of objects to be created and the difficulties in object identification.

On the other hand, other strategic analysis approaches apart from object orientation, especially soft methods (Rosenhead, 1989b) (see Chapter 2), can be used in an integrated manner to achieve the objective of problem understanding, resulting in object definitions required for subsequent decision analysis. At least, techniques from other strategic analysis methods can be applied to help complete and refine the definitions of decision entities obtained from their corresponding classes. Soft methods, which are becoming increasingly part of the operations research and management science field, are claimed to be able to facilitate individuals or groups to build up their understanding of a system in a structured and logical framework. They can facilitate the delimitation of the problem context for a decision problem and the understanding of the problem. For example, SSM (Soft Systems Methodology) (Checkland, 1989, 1990) offers a technique called cognitive mapping for behavioural representation for each stakeholder, which can diagrammatically illustrate the situation of a decision problem.

There are two basic ways to integrate soft methods into object orientation. The first is to use soft methods to organise various objects, which are initially generated from their existing classes, and to analyse their interactions with the methodology of soft methods. Objects are progressively defined during the analysis. The second way is to carry out the strategic analysis of a decision problem using soft methods first and then to utilise the output of the analysis as the basis to identify and define objects. The first way requires the existence of pre-created objects while the second may be strenuous when trying to identify objects hidden in the analysis output. For decision problems that have pre-defined classes, it is therefore advantageous to use the first way of soft method application while for other decision problems, soft methods may be helpful being used in the second way to identify objects and thereafter classes by generalising generic findings as decision classes. The application of soft methods in object orientation is a topic for future research.

After the definitions of some decision entities, including the entities of decision contexts and actors, and the initial consideration of decision elements, decision participants are able to brainstorm a decision problem based on the understanding resulted from strategic analysis. Under the guidance of facilitators and the help of domain experts, stakeholders identify various decision elements by using brainstorming techniques, reflecting their perceptions on a decision problem. Domain experts may need to check the identified decision elements mainly for their validity and completeness. Decision elements, such as action elements, decision alternatives and decision criteria, are then defined/refined to represent stakeholders' perceptions and expectation so as to make it possible to fulfil their aspirations after the decision problem is structured out of these decision elements. The next section discusses problem structuring from an object-oriented perspective.

7.4.3 Structuring

The process of structuring organises a decision problem into a structure that represents the problem in an analytic way for the purpose of problem exploring in finding out a solution for the problem at a later stage. Decision elements can represent the objective environmental components of the decision context and the subjective and context-dependent points of view. Decision elements are structured in such a way that the relations of these elements, such as influence relations, inclusion relations, hierarchical ordering relations, etc., and the value-systems of actors or stakeholders are made explicit. The conventional concept of problem structuring actually contains the three processes of decision making proposed for object-oriented decision analysis, i.e., initial understanding (IU), strategic analysis - brainstorming - decision element identification (SBI), and structuring. The first two processes deal with information collection, problem understanding, and the identification and definition of fundamental decision elements as the basis of problem definition, contained in conventional problem structuring methods (see Chapter 2). The third process, structuring, handles the final definition of a decision problem based on the definitions of fundamental decision elements made as a result of the first two processes.

The main tasks of structuring include the construction of decision criterion hierarchies and the generation of alternative sets, as indicated in Chapter 4 by the modelling of MCDM decision making procedures for natural resource management. The construction of decision criterion hierarchies is about organising the subjective and context-dependent points of view, i.e., concerns or objectives of stakeholders, into structures that indicate the hierarchical aspirations of stakeholders. The generation of alternative sets copes with generating possible plans of action, indicated as problem solution alternatives, in lieu of status quo along with their relations with the decision context, and forming alternative sets out of the generated alternatives, which present potential decision recommendations to decision participants.

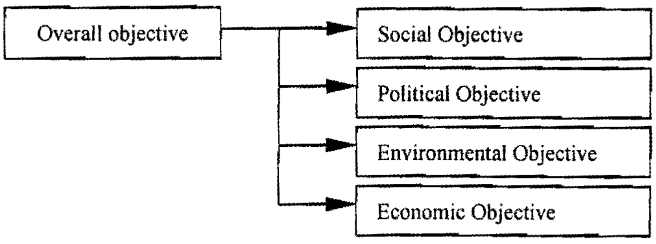
Decision alternatives for a decision problem and their relations with the decision context can feasibly be identified from the objects of decision elements and their interactions defined in the previous processes by means of instantiation and specialisation from their classes. Decision alternatives are generated in this way out of various decision action elements, which are the features or properties used to describe aspects of action plans for problem solution. *Knowledge relating to integration of the values of various decision action elements into alternatives may be maintained in the class of decision alternatives.* For example, some rules may be set up for the generation of decision alternative sets, which should be manageable in size, representative to represent all interest parties, and comprehensive so that virtually any decision alternative can be found within it possibly by means of interpolation between alternatives. The representation of this kind of sophisticated knowledge in objects is a challenging research topic. The relations of decision alternatives with the decision context are mainly included in the definitions of decision element objects, such as those of decision action elements, uncertainties, uncertainty dependencies, alternative constraints, etc. Their classes contain generic features that may be associated with decision action elements, which are the attributes of decision alternatives, while the pre-defined associations among these classes may provide direct clue to the relations between decision alternatives and the elements in the decision context.

Object-oriented decision analysis thus provides substantial promise as a mechanism for alternative generation, making use of the interactions between different decision element objects, such as actors, decision actions, and criteria. For example, an action by one actor concerned with some criteria may have impact on the actions of other actors, who are concerned with other criteria. A sequence of actions resulted from the interactions of different actors constitutes a decision alternative. A set of decision alternatives can be obtained by initiating various actions from different actors. In fact, a simple form of object interaction can assist the generation of decision alternatives. For example, decision alternatives can automatically be generated simply by combining cornerstone values defined in the individual objects of decision action elements. Each action element may have key values that are regarded be able to make significant difference to the consequence of the decision to be made. The combinations of different values from various action elements result in some representative decision alternatives for the decision problem under consideration. Decision alternatives, as another example, can easily be auto-screened out by the rules or/and thresholds embodied in the objects as well. Some ranges of values or options of a certain action element may not be permitted according to some governmental rules, or are not feasible in the practical realisation of certain actions because of different causes such as resource limitations and obvious disastrous consequences agreed by all decision participants. It is noticed that a DSS can be of great help to decision alternative generation in object-oriented decision analysis since the system allows objects to interact with themselves as auto-running computer codes. Decision alternative generation by object orientation is a very important topic for the future research.

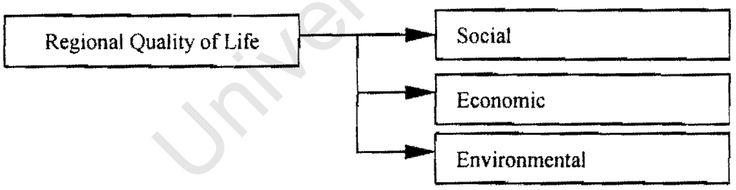
Criterion hierarchies are constructed out of the criteria, which are brainstormed and defined under the guidance of their classes in the previous process, to enable stakeholders to distinguish preferentially between the alternatives under consideration. Knowledge relating to construction of decision criterion hierarchies may also be kept in classes. For example, the decision criterion hierarchy at the overall level of a decision problem should include all the stakeholders in the problem. Criteria at a lower level of a hierarchy should be more specific than those at a higher level, while the lowest criteria need to be operationally meaningful so that a decision alternative can be more or less unambiguously judged accordingly.

Example:

A class of overall criterion hierarchies is defined as follows. As discussed in Chapter 4, for natural resource management problems, influence factors, which may be built into decision criteria for systematic analysis at a later stage, are roughly classified into four categories: social, economic, political, and environmental factors. For each of the influence factors, there are still more subclasses of factors to be considered.



The actual overall criterion hierarchy for the decision problem is generated under the guidance of the class. Consideration is also given to the specific context, as shown in the following diagram for the example decision problem. In this case, political objective is omitted because there appear to be no strong political conflicts in the problem and implied political influences are contained in the other three types of objectives.



Low level criterion hierarchies are more complicated than the overall hierarchy. But the knowledge for their construction can be represented to some extent in relevant classes, which are be able to guide the creation of these individual low level criterion hierarchies.

Object-oriented MCDM thus allows a decision problem to be structured based on "objects". Decision alternatives and criterion hierarchies are "objects" themselves and are constructed out of other "objects" and their interactions. The construction may be facilitated by the reuse of knowledge represented in classes.

The structuring process is mainly concerned with the generation of decision alternative sets out of existing relevant decision elements, and the organisation of decision criteria, which are also defined in the previous processes. During the structuring process, the generated decision alternative sets and the constructed decision criterion hierarchies may be checked by domain experts and facilitators mainly for their validity and completeness. Decision making may return to the previous stages for more information if needs be (see Chapter 4 for the discussion of the sequences of decision making activities). The next section discusses the phase of exploring.

7.4.4 Exploring

Based on the structured definition of a decision problem, the process of exploring explores the profiles of various decision alternatives by various means. A decision problem may be analysed and understood during the processes of identifying and defining various decision elements, such as criteria, action elements, and alternatives. A solution may be recommended after observing the achievements of alternatives resulted from subjective judgements, individual preferences, prospective impacts on the decision context caused by the a suggested decision alternative, and actual effects on the society resulted from the implementation of a decision.

The process of exploring actually contains the phases of evaluation, choice, and implementation included in the conventional decision making process models discussed in Chapter 2. After the generation of decision alternative sets and the construction of decision criterion hierarchies, alternatives can be concurrently compared by individual stakeholders and domain experts according to their criterion hierarchies. At the same time, criteria can also be evaluated. A choice is made out of the evaluation of alternatives before a possible planning for the implementation of the decision made.

During the phase of evaluation, many decision making activities are carried out, including the elicitation of subjective judgements of stakeholders, value functions for ranking alternatives, value weights for measuring the trade-offs amongst criteria, and risk preferences, uncertainties exploration, performance of necessary statistical analyses, and conduction of logic-based reasoning. The assessment of the consequences of each alternative may involve the use of a variety of models, e.g. economic models and hydrological models, and other traditional methods of analysis, such as environmental and social impact studies and cost benefit analyses. Alternatives can be compared at different levels of goal achievement indicated in a decision criterion hierarchy, and decision participants can assess alternatives holistically from the point of view of their own interests.

The evaluation is aggregated to obtain a choice once preferences to alternatives and weights to all the criteria are assigned. Aggregation can be carried out at different levels of the hierarchy according to an achievement measure. The expected achievement of each alternative is calculated. Results of various kinds, including a chosen alternative, alternative categories, and ranked alternatives, are available to decision participants. Quantitative comparisons may be usually made for the direct ranking of the expected achievements for decision alternatives on all of the objectives. Other kinds of comparison of decision alternatives, e.g., on any two selected criteria, are also useful to enable substantial satisfaction on the preferred alternatives by the stakeholders. The sensitivity of the weighted achievement to key probabilities, weights, risk preference parameters, and critical variables is analysed.

Implementation of the decision made may be planned by relevant agents with possible participation of decision participants. Implementation agents establish a schedule of tasks that must be completed to fulfil the desired aspirations of stakeholders. Critical activities and resources may be identified for the implementation. Necessary modifications may need to be made during the implementation procedure as well as drastic rethinking of the implementation plan and possible re-initialisation of the whole decision procedure. The implementation planning is mainly concerned with assessing the execution of a decision. More insights may be gained from the planning and from the possible action taken thereafter.

The exploring might lead to new questions that were either not fully examined or not even considered during the previous phases of decision making. Due to the implementation planning and other decision exploring activities described above, insights are obtained at various points about a decision problem and its decision context. New findings in the analysis phase may undoubtedly generate new insights into a decision problem. There might be a need for revisions to the problem definitions and decision recommendations. The decision making procedure may go back to the previous stages to reiterate some processes, as discussed in Chapter 4. A new round of decision making may need to take place.

Once again, object orientation offers a mechanism to facilitate decision exploring. As discussed in Chapter 6, the activities of decision making included in this phase can be naturally represented by the behaviour, which is a set of operations, and the interactions of objects. Operations of objects describe behaviour of classes, which mainly are decision making actions. Objects interact with each other by sending and receiving messages in order to initiate various decision making activities. Object interactions, represented by directed control flow and messages transmission between objects, describe decision making paths for various decision participants. Most importantly, operations and interactions are intrinsic features of objects. *Decision making activities can be carried out by these objects on their own in a DSS, which is composed of various objects.* There is no need for additional definition of decision making functions in such a system.

In the DSS developed for the water resource management in South Africa (refer to Chapter 8), decision exploration is carried out as a result of object definitions and object interactions. The MCDM method used in the DSS was based on MAVT (Multi-Attribute Value Theory), which was the approach favoured by the facilitator in this case. After the construction of the decision problem, criteria are evaluated quantitatively or judgementally. Alternatives are compared according to the criterion hierarchies constructed. Their scores are then aggregated at different levels of the hierarchy according to a value function. The final aggregation produces the uppermost level of aggregation scores, representing the ordering of the alternatives to decision participants. The sensitivity of the preference ordering to the underlying weights and values is then analysed. Implementation of the decision may also be planned.

The decision making tasks in this phase can be viewed as carried out by "objects", which then provide the system building material for a DSS, and the interactions between them. These "objects" and their interactions relate naturally to the system functions of decision making in a DSS. Various "classes" can then also be reused in DSS development (refer to Chapters 3, 4 and 5).

Actually, once the decision problem is fully structured, its exploration is essentially a computational task. It is the formulation of a decision problem which is primarily a conceptual and representational issue. DSSs that are implemented to support object orientation can play an important role in the process of exploring since objects and their interactions can carry out the system behaviour of such a DSS. The next section discusses the relationship of DSSs and object-oriented decision analysis.

7.5 Decision Analysis and DSSs

Object orientation provides a uniform methodology to model decision problems across the entire procedure of decision analysis, and DSS development. As shown in Chapter 3, decision analysis and DSS development can be carried out in an integrative way based on object orientation. In this approach, decision problems are represented and defined with objects of decision entities, which can be created from existing corresponding classes. A decision problem may then be analysed based on defined classes and with the assistance of strategic analysis. Various decision making activities included in problem understanding, structuring and exploring are then implemented via the behaviour and interactions of different objects of decision entities. At the same time, the concepts of object orientation, such as objects and object interactions, are in this way also used as fundamental tools in the analysis, design, and implementation of object-oriented DSSs. The identification of the objects for decision making processes, the description of how objects interact, and the analysis of their principle attributes and operations constitute foundations for the analysis and design of DSSs. The functions of DSSs are accomplished by the interactions and the behaviour of individual objects.

On the one hand, DSSs facilitate object-oriented decision analysis, mainly in two ways. The first is via the support of DSSs in decision making tasks. DSSs are supposed to support some or all of the decision making processes for solving a decision problem. As shown in Chapter 4, a DSS as usually conceived is a micro-system that comprises entities to facilitate some decision making processes in the macro-system, which is an integral conceptual framework for understanding and analysing multicriteria decision problems, the decision making procedure and DSSs. The macro system is a system of general perception, dedicated to solve natural resource management problems by using all means of problem analysis, such as workshops and interviews, to find a satisfactory policy alternative for all the parties involved. The second way of DSS facilitation in object-oriented decision analysis is through the power of computers. Computerised DSSs bring about the convenience of manipulating the concepts of object orientation, such as object interactions and class maintenance. Firstly, by using the power of computer systems, classes of decision analysis kept in a DSS can easily generate individual instances (objects) in the case of class instantiation, and can also be easily specialised in the case of class inheritance. Class maintenance, such as modification, addition and deletion, can be implemented with ease. Secondly, objects stored in a DSS are executable problem modules that can activate an action or be activated upon receiving a message from other objects. Interactions of objects are automatically carried out by message transmission among objects. Thirdly, as discussed in the above section, computerised DSSs are also able to take over human's burden of representational and computational tasks in formulating a decision problem and exploring a formulated decision problem. DSSs are an efficient way to implement object-oriented decision analysis.

On the other hand, as discussed in Chapters 3, in the framework of object-oriented decision support, modelling of decision making provides the foundation for the modelling of DSSs. As shown in Chapter 4, the object-oriented modelling of MCDM decision making processes facilitates the analysis of DSSs for MCDM in natural resource management. Modelling of decision making can identify the essential and fundamental classes and class relationships needed for DSS modelling and development. *Real world objects in natural resource management problems by MCDM are generalised into classes on the basis of common properties. These classes are easily re-used in the development of DSSs.* Classes and class relationship hierarchies offer different points of view for decision problems and DSSs, and also provide an easy way to allow communication between decision analysts, stakeholders, domain experts, and DSS researcher in a simple and transparent way. Reusable classes allow a DSS model to be reused efficiently to construct a specific DSS by instantiating existing classes and linking the objects obtained together in some way.

A conceptual framework composed of classes and their interactions provides fundamental specifications for DSSs as well as decision making paths for decision participants. By integrating all classes and class relationships into a single diagram, a general class framework for decision analysis and the development of DSSs for MCDM in natural resource management is obtained as shown in Chapter 6. The functions of decision analysis and the behaviour of DSSs can be integrally explored in this way. The next section summarises the advantages of the application of the object-oriented philosophy in the field of decision analysis.

7.6 Summary of Advantages of Object-Oriented Decision Analysis

The fundamental contribution from object orientation to decision analysis is that object orientation offers a uniform philosophy and methodology for decision analysis and DSS modelling in a unified context. As discussed in Section 7.2, object orientation allows the world organised around independent yet interacting objects, which possess their own attributes and behaviours. The world itself and any part of it are composed of physical or non-physical objects constructed around concepts instead of their functionality. Decision problems, decision making processes, DSSs, and their context can all be naturally modelled by a collection of objects in a simple and transparent way that is focused on the conceptual side instead of the functional side of the world.

Secondly, the methodology of object-oriented decision analysis is an important bridge between analysts, stakeholders, other decision participants, and DSS researchers. Object orientation allows a simple concept, which is "objects", to be used to explore decision problems and DSSs. Objects and their various representations are a device that can readily be sketched by a layman and yet be so carefully defined that they can be easily implemented in computer systems. Object-oriented decision analysis crosses the border between the graphic view of relationships that is very convenient for human beings and the explicit equations and numbers that are the province of present computers. In addition, each party involved in the decision making processes is modelled as an object in object-oriented decision analysis. Their communications and interactions are represented as messages transferred between the objects. This makes the methodology sound for applications to group decision making with geographically and culturally dispersed individuals.

Thirdly, object-oriented decision analysis offers an easy way for the understanding of decision problems, decision making and eventually the functions of DSSs. Each object or clusters of objects can represent a different part of a decision problem. Different viewpoints of decision problems and decision making paths can be observed from various hierarchies and interactions of objects. Reusable classes that have been tested in the field of similar decision problems offer reliable understanding of relevant basic entities, which may be generated from existing classes, resulting in a good understanding of the primary components of a problem. Prototypes for a specific decision problem can be obtained respectively by instantiating a constructed problem definition. The obtained prototypes may help understand a decision problem, getting decision participants a basic knowledge of decision making functions, which are implemented in a DSS, and lead to high productivity and high quality of decision analysis and DSS development.

Furthermore, object orientation offers a mechanism to manage complexity in decision analysis. An object is a complete entity that encapsulates different properties of attributes and behaviours performing definite tasks for a decision problem and containing all components needed to carry out the tasks. Object encapsulation allows information hiding to defer the definition of internal detailed activities of an object until necessary. In addition, class abstraction and class inheritance simplify a decision problem by reducing the number of independent components at the initial stages of analysis so that rapid initial understanding can be generated. Besides, class relationships may help organise a decision problem in a hierarchical way offering different points of view of the problem to various actors. A decision problem partitioned on the basis of individual objects help with scalability which can manage complex large problems through scaling up from small to large.

Finally, object-oriented decision analysis can improve the efficiency and effectiveness of decision analysis. The use of existing knowledge and past experiences is allowed by utilising the classes of the generic aspects of decision problems and decision analysis. The problem context, people involved, decision elements, etc, can be reused for future decision problems with similar features. Through reusing existing information and products, costs are decreased and accuracy is increased. In addition, object orientation provides a uniform tool to deal with almost all the aspects of decision analysis. People do not need to piece together a patchwork quilt of tools to deal with various phases as they arise over the life of a project. Object orientation will be able to allow these phases to be carried out in a uniform and coherent way.

There are many other contributions available from object orientation to decision analysis, as discussed in the previous chapters. There are also some potential benefits that object orientation can bring to decision analysis. For example, flexibility of decision analysis needs to be explored in order to provide flexible ways for decision analysis. Object orientation allows bottom-up and top-down ways of analysis, partitioning an overall problem object into component objects and integrating objects of individual entities into a whole object. As discussed briefly in Section 7.4, other methods of soft problem analysis can be embedded in strategic problem analysis with an option to keep object orientation as the analysis basis.

7.7 Conclusions

Object orientation offers a uniform model for the modelling of decision problems, decision making procedures, and DSSs, and brings about a communication mechanism for various actors, understanding of issues related to the solution of decision problems, management of complexity, and effectiveness and efficiency of decision analysis and DSS development.

The object-oriented methodology of decision analysis meets the requirements by problem analysis and structuring, which is the most critical phase in decision making processes as discussed in Chapter 2. The methodologically and philosophically sound methodology can naturally model the real world from different viewpoints in a simple, transparent and flexible way, and is able to reuse the past experiences and relevant knowledge. Soft and hard problem analysis approaches can also be embedded in the methodology. The four main streams of problem structuring thoughts proposed by (Woolley and Pidd, 1981) are also comprised.

There is also a strong theoretical foundation for object orientation to model decision making activities and DSSs. Various entities in a decision making procedure can be represented as objects. Objects and their interactions carry out the activities of decision making as well as the major functions of a DSS. Operations of objects describe some basic decision making actions while object interactions describe decision making paths for various decision participants. These objects and interactions further model the system behaviour of DSSs.

The main objective of this chapter has been to describe the methodological guidelines for object-oriented MCDM in natural resource management by presenting a general framework of object-oriented decision analysis, which shows the basic ways and general processes to utilise object orientation in decision analysis. The fundamental ideas behind this framework include the object-oriented philosophy to model the world and reuse of existing classes.

There are four processes, i.e., initial understanding (IU), strategic analysis - brainstorming - decision element identification (SBI), structuring, and exploring, of decision analysis included in the decision making process model of object-oriented decision analysis. In the phase of initial understanding, the decision context, actors, and other decision entities are identified by means of instantiation of relevant classes which are generated beforehand out of relevant knowledge and past experiences for a problem domain. Initial understanding of a decision problem may be achieved as a result of the preliminary analysis of these objects.

The phase of strategic analysis - brainstorming - decision element identification (SBI) is a further phase to understand the decision problem under consideration by systematically considering all the objects involved. The basic skills of object-oriented strategic analysis are demonstrated in a form of actor-oriented object analysis. Other techniques of strategic analysis such as soft methods can also be utilised as supplementary approaches to obtain a thorough understanding of a decision problem. On the basis of decision element objects created from their classes, brainstorming techniques are then applied to produce a complete and refined collection of decision elements for the structuring of the problem. It is also noted that pre-existence of classes is not a requisite for object-oriented decision analysis.

In the process of structuring, a decision problem is structured out of various decision elements with the guidance of problem structuring knowledge stored in relevant classes. Decision elements are structured in such a way that the relations of these elements, such as influence relations, inclusion relations, hierarchical ordering relations, etc., and the value-systems of actors or stakeholders are made explicit. The tasks of structuring mainly include the construction of decision criterion hierarchies and the generation of decision alternative sets.

In the process of exploring, a structured decision problem is explored for the profiles of various decision alternatives. This process includes elicitation of preferences and judgements of decision participants, studies of various outcomes of selection by values or by actual impact on interest parties involved, implementation planning, or even partial implementation of the decision made. A solution may be recommended after observing the achievements of alternatives resulted from the exploration. The phase of exploring in object-oriented decision analysis actually contains the processes of evaluation, choice and implementation included in some decision making process models reviewed in Chapter 2.

The role of DSSs in object-oriented decision analysis was noted as very important mainly due to the convenience brought about by computerised systems in manipulating some concepts of object orientation. Besides, all decision making activities contained in the decision making processes are carried out by the operations of various objects and their interactions, which can be easily implemented in a computerised DSS. DSSs are an efficient way to implement object-oriented decision analysis. Object-oriented decision analysis usually needs the assistance of a DSS. On the other hand, the object-oriented modelling of decision analysis constitutes a basis for the modelling of DSSs.

Though some specific issues in the whole decision making procedure of object-oriented decision analysis still need further research to make use of the full potential of object orientation, the potential contribution of the object-oriented philosophy to decision support appears well-established. It offers a philosophy and methodology for decision analysis and DSS modelling in a unified context. As an important mechanism to bridge the gap between analysts, stakeholders, other decision participants, and DSS researchers, object-oriented decision analysis can manage problem complexity, and help for the understanding of decision problems, decision making and eventually the functions of DSSs, leading to improved efficiency and effectiveness of decision analysis. The next chapter demonstrates the practical implementation of the object-oriented approach to DSS development and decision analysis for natural resource management.

Chapter 8

Practical Implementation of the Object-Oriented Approach

8.1 Introduction

Based on the modelling of decision problems, decision making activities, and DSSs in a uniform way, the object-oriented approach to decision support provides a model for DSS development and a practical methodology for analysing MCDM problems of natural resource management. It facilitates the reuse of past experiences and relevant knowledge, communication, management of complexity, and the understanding of the issues related to decision problems, decision making and eventually the functions of DSSs, leading to more effective and efficient decision analysis and DSS development.

The objective of this chapter is to demonstrate the practical implementation of the object-oriented approach in the development of a specific DSS and in the decision analysis of MCDM natural resource management problems. The modelling results of DSSs are utilised in the design of specific systems. The DSS model defined in the study plays a very important role in the system development of a specific system. Reusable outcomes of analysis and design can speed the development of the specific system. This is further shown by the development of a practical system termed here the WRC DSS as it was developed as part of a project for the South African Water Research Commission (WRC).

The system, WRC DSS, has been made available to public access through the Department of Statistical Sciences at the University of Cape Town. At the time of the writing of the thesis, the system is being transferred to a new web site, whose address will be supplied on request.

Then the general framework of object-oriented decision analysis, which includes four processes of decision making as discussed in Chapter 7, is applied to a hypothetical decision problem case. The methodology is shown to be effective and efficient in structuring and analysing the problem.

This chapter is organised as follows: Section 8.2 discusses the use of the DSS model, the implementation of specific DSSs, and the development of WRC DSS. Section 8.3 demonstrates the practical application of the object-oriented approach in the removal of alien vegetation from Table Mountain. Three workshops were carried out to evaluate the performances of the system and the decision analysis methodology. Section 8.4 analyses the questionnaires from the workshops. Conclusions are contained in Section 8.5.

8.2 The DSS Model and DSS Development

A DSS model mainly includes the outcomes of DSS domain analysis, i.e. the general DSS requirements, various classes and their relationships, design of subsystems and system architecture, other documentation, relevant diagrams, etc. As reviewed in Chapter 2, there is a great need for modelling of DSSs to assist the development of DSSs for natural resource management. A DSS model is the mechanism to bridge the gap between the DSS researchers and decision analysis practitioners as it enables the DSS development and evaluation understandable to decision practitioners and DSS researcher alike. A DSS model can deal with the complexity of DSS development and make the development of a specific DSS affordable in terms of both time and cost. Ideally the DSS modelling methodology should be integrated with that for modelling decision problems to improve the effectiveness and efficiency in both aspects. Relevant entities of DSSs should be modelled in a simple and transparent way. An ideal DSS model should support group decision analysis for all the phases of decision making.

In this study, DSS modelling is carried out along with the modelling of decision making as discussed in Chapters 3 and 4. General DSS requirements were captured, and classes were identified in Chapter 5. Class interactions and relationships were analysed in Chapter 6. In this section, DSS subsystems and system architecture are designed first. DSS implementation is then discussed.

8.2.1 Subsystems and System Architecture

The primary classes are organised into subsystems, which are then deployed in a system architecture before planning system implementation. A subsystem is a collection of classes that performs certain functions. A system architecture shows a conceptual system framework about the physical deployment of the subsystems and other major components.

The subsystems of the DSS model are shown in Figure 8.1. The subsystem of system administration administers the users and monitors the system. The problem understanding subsystem helps users understand the problem context and the decision making processes. The problem structuring subsystem deals with identification of basic decision elements, such as attributes, criteria and uncertainties, during the brainstorming phase, the generation of alternatives, and the construction of criterion hierarchies. The evaluation subsystem elicits stakeholders' judgements and finally aggregates evaluations to obtain a result. The database subsystem manages documents to supply data and information for other subsystems. The user interface subsystem handles user commands and communications between the machine and the users. User interface is no longer a major concern in the modern software engineering technology even though it is still an important issue. The requirements of most user interfaces can be easily met by many commercial software development tools.

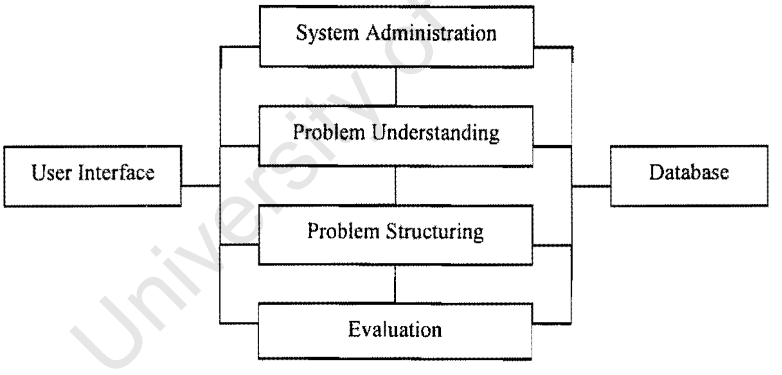


Figure 8.1: Subsystems

The architecture diagram for the DSS model is shown in Figure 8.2. It shows the physical computing units in the system, their devices, and the communication links between the computing units. Computing units are capable of executing programs while devices can only support the execution of the programs. Subsystems of system administration, problem understanding, problem structuring and evaluation reside in the servers. They are accessed through the network access tools and operated via the user interface subsystem. To be precise, their codes are kept in the servers while they are operational on the front-end machines. They can be downloaded or accessed via other means by client machines while databases are stored and can also be administrated in the servers.

The network, servers and front-end client machines are the three main hardware components in the system. A server is a machine or process that provides a service to another process. Typically, servers are computers that exist on a network configured to provide a particular resource, such as files, data, Web pages, or application processing services. A network is a collection of network interface cards, hubs, routers, and wires tied together to form a group of interconnected computers. The network is the media that connects various users of the system. It provides a common communications mechanism that exists between users and different devices. Front-end client machines are normally local desktop machines, either personal or workstation computers, which have the capacity to access a network.

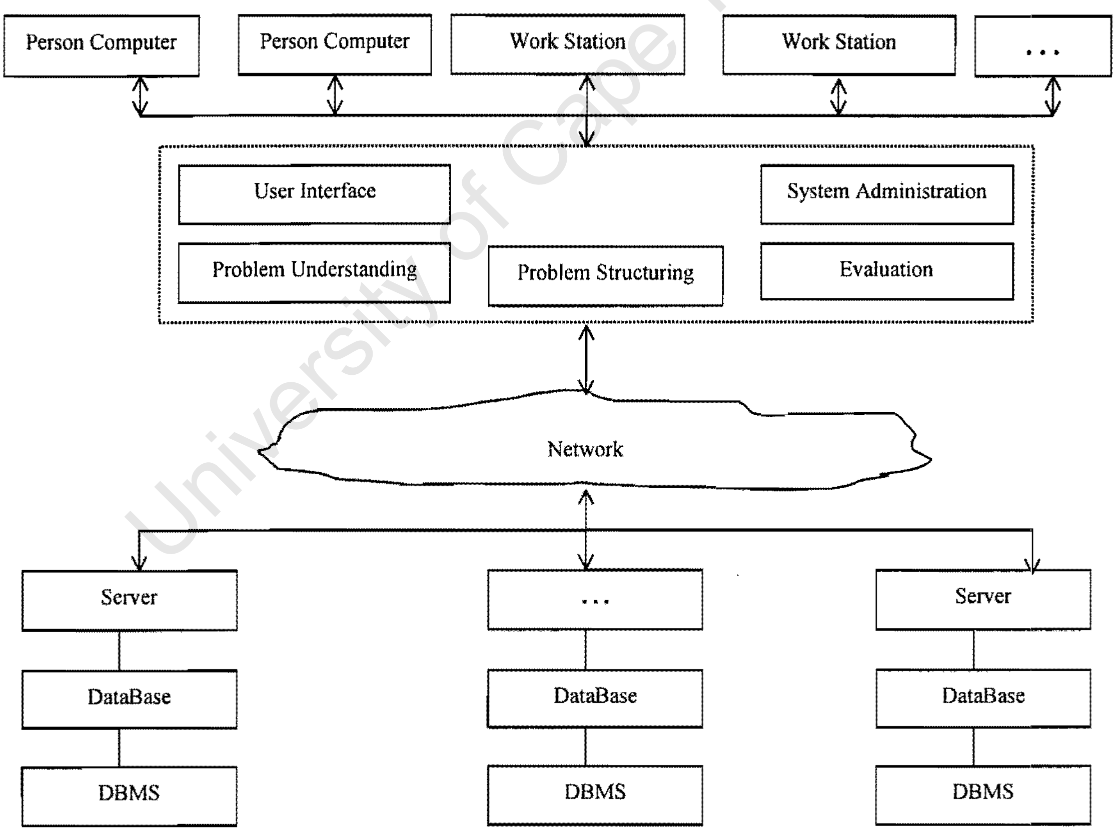


Figure 8.2: System Architecture

Stakeholders and other system users access the system through the network without any constraints of geographical locations. Stakeholders of a natural resource management problem may be geographically dispersed across a large area and even internationally. They need to be able to use a DSS whenever and wherever they want to. This can be achieved by the accessibility of network nodes in offices and homes and via wireless communication means. The access to the network ensures the availability of a DSS. This is one of the basic differences between traditional and Internet DSSs.

8.2.2 System Implementation

Various aspects of the technical DSS evaluation principles discussed in Chapter 2 are considered for the implementation of a specific DSS in order to fulfil the main desirable DSS features defined by these evaluation principles. In Chapter 5, some technical evaluation principles for system implementation, such as data interfaces, demonstration examples, and system status displaying, were discussed. The remaining DSS evaluation principles are mostly technical details that vary from system to system and will change over the time with the advancement of computer technology. However, these evaluation principles, including user friendliness, data processing, and system installation, should be taken into account at the stage of design of a specific DSS. Refer to Chapter 2 for more description.

Table 8.1: DSS Implementation Checklist

Item Name	Main Consideration Issues
General Implementation Consideration	
Development Environment	Development platform; Programming Languages
User interface	User interface type; Implementation tools
Learning and Ease of Use	Operations; Examples
Help Information	Live tutorial; Help display; Help availability
Data Processing	Data base management; Data format
Interface to other systems	Application interfaces
System Installation	Installation process; Equipment requirements
Class library	Class library features
Decision making Consideration	
Problem Explanation	Live demonstration; Content display
Brainstorming	Communication; Elicitation techniques
MCDM Method	MCDM method selection
Uncertainty Analysis	Uncertainty resolution approach
Decision Elements	Value display and representation

A checklist of main items of considerations is provided in Table 8.1 for the examination of the technical system requirements. The decisions made with regards to these items may have an impact on the detailed design and implementation of a system. There are two broad categories of considerations. The first category is about general implementation considerations, which are primarily based on the DSS evaluation principles for technical implementation. The second category deals with decision making considerations, which are mainly related to specific decision making functions in real cases.

The general considerations for the implementation of a specific system in fact contain all the aspects in the DSS evaluation principles for technical system implementation. Development environment is mainly concerned with the programming languages used to code the system and the development platform in which the system is developed. User interface primarily deals with the user interface implementation tools and the type of interfaces, i.e., window based or command based. The main concerns of learning and ease of use include ease of systems operations and demonstrations of problem examples. Help information can be in different forms such as live tutorials, which can be either just animation (moving graphs) or multimedia (audio-video display), and textual and graphical display. Data processing mainly deals with data base management, and data input and output format. Another concern of data processing is the limits to the problem size a DSS is able to cope with. Effort is made to improve the efficiency of data processing, algorithm design and computer resource allocation so as to limit this constraint. The application interface to other systems needs to find out which data or file is needed to import or export to other systems, what are these systems, and what are the formats for the data and files. System installation considers the installation process and equipment requirements by the system. The class library is concerned with the selection of class libraries, which are organised collections of classes.

Class libraries are needed for the analysis and design of DSSs. It is advantageous to make use of the existing classes of system building material, such as decision elements and other system components identified in the DSS model. Reuse of the existing classes will speed up the development of DSSs. Many commercial libraries are on the market nowadays for everything from user interface utilities, communication mechanisms, to libraries geared towards specific application domains. But unfortunately there is not such a class library for the analysis and design of DSSs.

In future, commercial class libraries for DSS development might be available to DSS developers when more attention is paid to their reuse in the area of DSS development.

The decision making considerations for the implementation of a specific system involve the implementation of major dynamic decision elements and some other decision making tasks, such as problem explanation, brainstorming, and MCDM methods. Problem explanation allows users to understand the decision problem by using some approaches including animation (moving graphs), multimedia (audio-video display), textual, and graphical forms. Brainstorming is based on communications among users and elicitation techniques. Mechanisms of communication may include one-to-one (private communication) and broadcast. There is also a need for the selection of elicitation techniques and the consideration of a textual or graphical way to display inputs. MCDM methods need to be chosen as the basis of decision analysis in the system. The MCDM method chosen decides the manner in which the alternatives need to be evaluated, and the mechanism by which satisfaction of the aspirations or desires of stakeholders are measured. For uncertainty analysis, there is a need to find out what approach is used to resolve uncertainties in the system. A decision needs to be made about displaying the content of most decision elements, whether value, relationship, dependency, constraint, rule, or model,. Textual or/and graphical display modes are used as basic approaches for the input and output of various contents.

In addition to the considerations of these technical aspects, classes and their relationships identified during the analysis phase are the primary input at the system implementation. Additional implementation-oriented classes are discovered together with their attributes, operations, and relationships. Classes are then implemented with specific programming languages.

8.2.3 Model Utilisation in the Development of DSSs

The DSS model built in the study provides a general framework for the analysis and design of DSSs. It also allows such a system to meet the particular requirements of the evaluation principles for DSSs of MCDM in natural resource management. All entities of decision problems, decision elements, and system building components are represented in the DSS models as classes. Diagrams are used to illustrate the concepts of the model. The interactions among various classes accomplish the functionality of decision making. A DSS developed on the basis of this DSS model will be able to support the decision making processes of a group of participants. In addition, such a system can meet the evaluation principles for DSSs since these principles are used as a guide when modelling DSSs. Various people, including stakeholders, facilitator/analyst, and developers, can understand not only DSSs but also decision problems and their solving processes in a simple and transparent way. Moreover, the DSS model helps determine a system architecture for the design of a specific system.

There are three basic methods, i.e., elaboration, translation, and copying, to utilise the DSS model. Elaboration successively refines the elements of the DSS model to cater for detailed requirements of a specific system. Analysis details are added based on the decisions to be made about MCDM methods, elicitation techniques, uncertainty solution approaches, value comparison ways, reporting forms, etc. Translation provides a means of transformation of relevant entities between the DSS model and the actual system being developed. Some aspects of the DSS model may need minor changes such as a specific name for decision alternative (e.g. policy scenario). For some systems, analysis and design can also be carried out by simple copying. Copying allows direct borrowing of system entities and removing those unnecessary or impractical aspects from the DSS model. The mechanisms of elaboration, translation, and copying, are utilised to make use of the DSS model to obtain reusable analysis and design outputs.

The development of a system discussed in the next subsection shows how the DSS model and the checklist of Table 8.1 was used in the development of DSSs by checking the items in the checklist and also by using some combination of the model utilisation methods.

8.2.4 The Development of WRC DSS

A DSS for water resources management in South Africa was developed under contract to the South African Water Research Commission (WRC) by using the existing DSS model and the technical implementation checklist discussed above. The objective of this WRC DSS is to support the process of applying MCDM concepts to decision making for water resource management in South Africa. The system facilitates managerial decision making by providing tools, procedures, and data that add structure to the decision making processes. The MCDM method used in WRC DSS is a MAVT (Multi-Attribute Value Theory) based approach called Scenario-Based Policy Planning (SBPP) (Stewart, Scott, and Joubert, 1993; Stewart and Scott,1995, Stewart, Joubert, Scott, and Low, 1996).

WRC DSS is a group DSS (GDSS) based on the Internet. Decision making can be carried out without geographical restriction (users need only Internet access). The system allows a group of decision makers working together as a team to share information interactively, generate ideas and actions, choose alternatives and negotiate solutions. Interactive information appears in different forms, including the most commonly used web page formats, such as HTML (Hypertext Markup Language) pages, to guide the entire procedure of system operations. Users need only follow the flow of information from the Internet in order to fully make use of the system. Figure 8.3 shows the home page of the system under Internet Explorer (IE) 5.0.

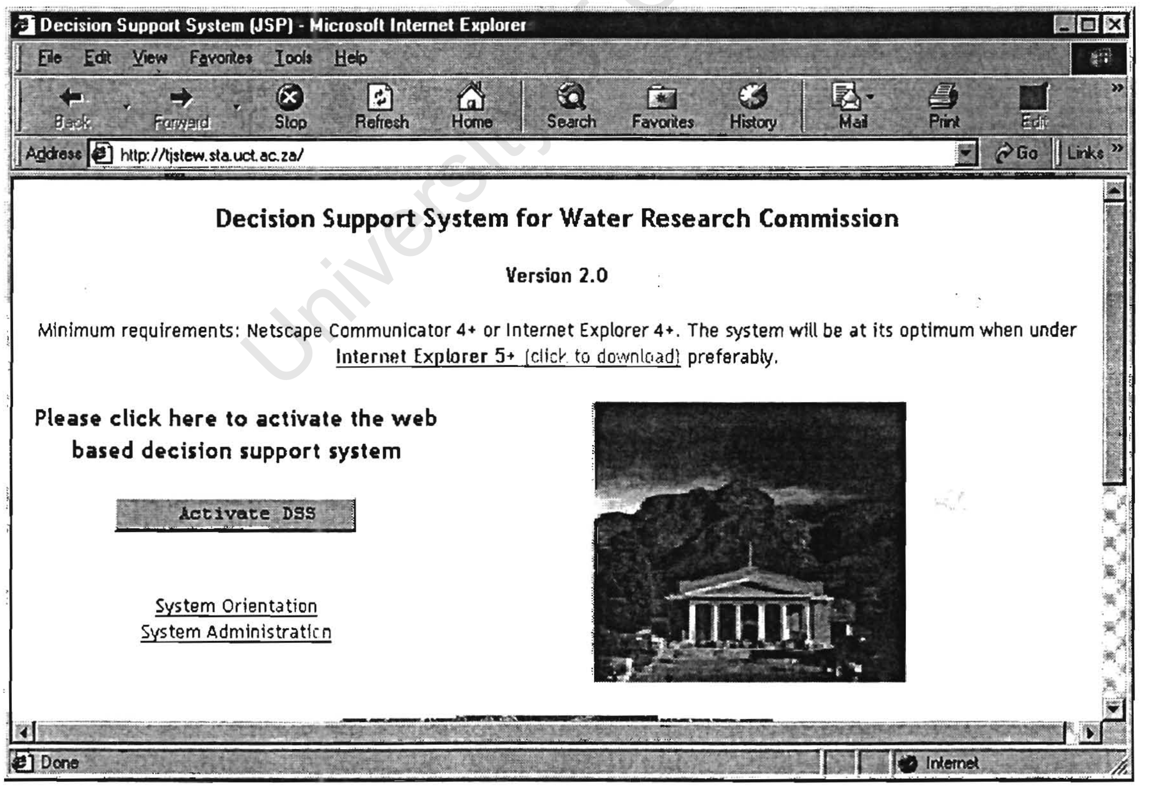


Figure 8.3: The Home Page of WRC DSS

WRC DSS supports most of the decision making processes, which have often been described as consisting of several distinct and iterative stages: problem structuring, evaluation, aggregation and implementation (see Chapter 2). It facilitates all processes except implementation, which mainly concerns the planning of tasks to implement the decision made. At the phase of problem structuring, the problem under consideration is identified and defined in terms of criteria, alternatives, and other related data. These may be thought of as the modules of problem structuring. Figure 8.4 shows the identification module and Figure 8.5 demonstrates the value tree construction module.

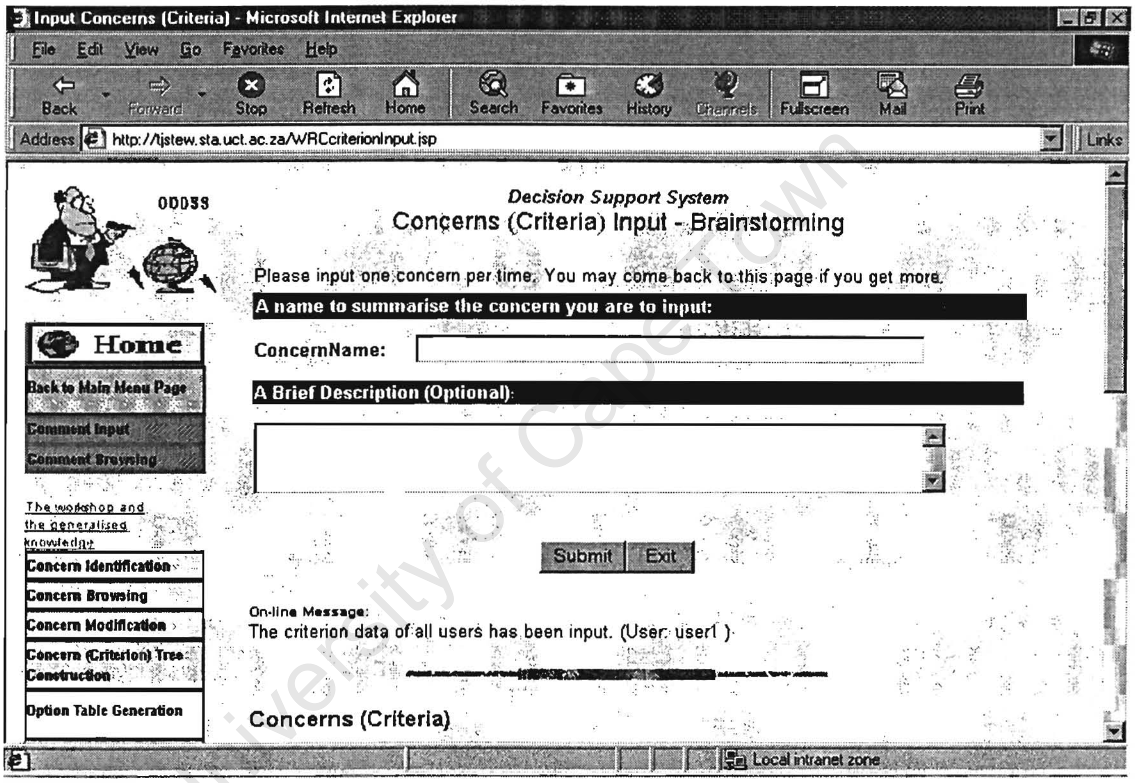


Figure 8.4: Identification of Concerns or Criteria

The evaluation and aggregation phases include the module to elicit subjective judgements or value functions for evaluating alternatives, and the module to elicit weights for measuring the trade-offs amongst criteria. They also calculate the weighted value of each alternative. Finally, the sensitivity of the weighted value to weights is examined. Figure 8.6 shows the module of decision alternative evaluation. Figure 8.7 demonstrates the module of aggregation together with some forms of result presentation.

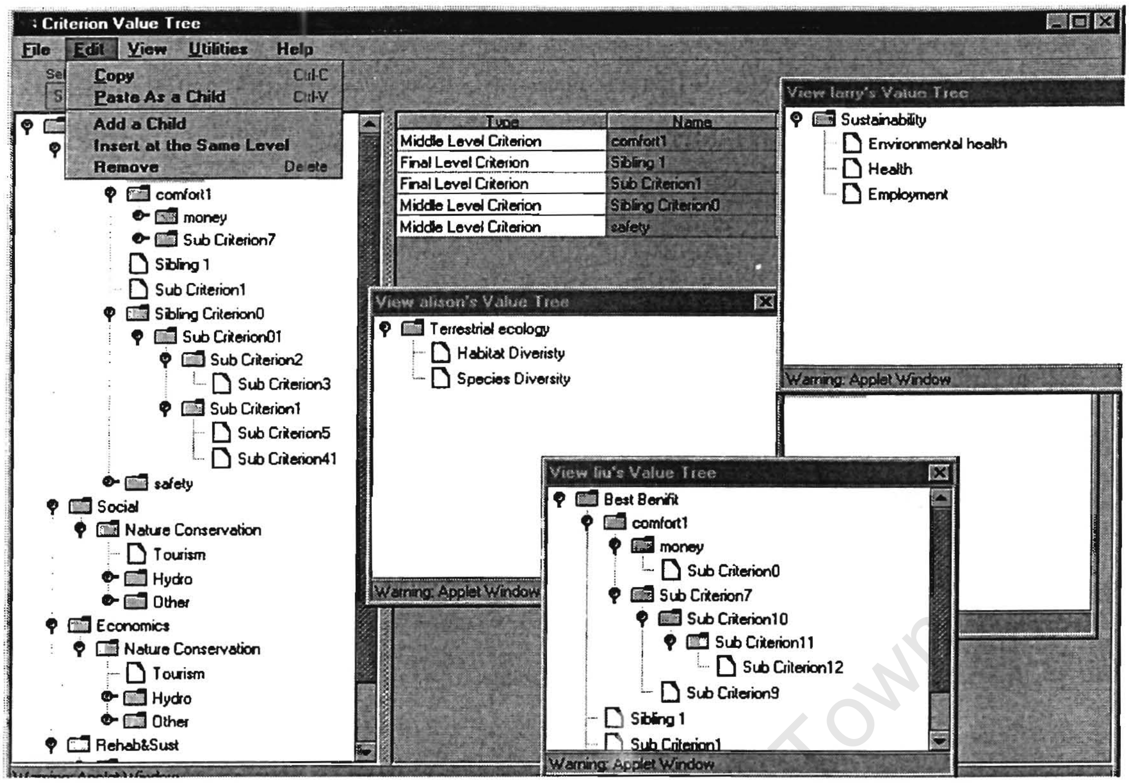


Figure 8.5: Construction of a Value Tree

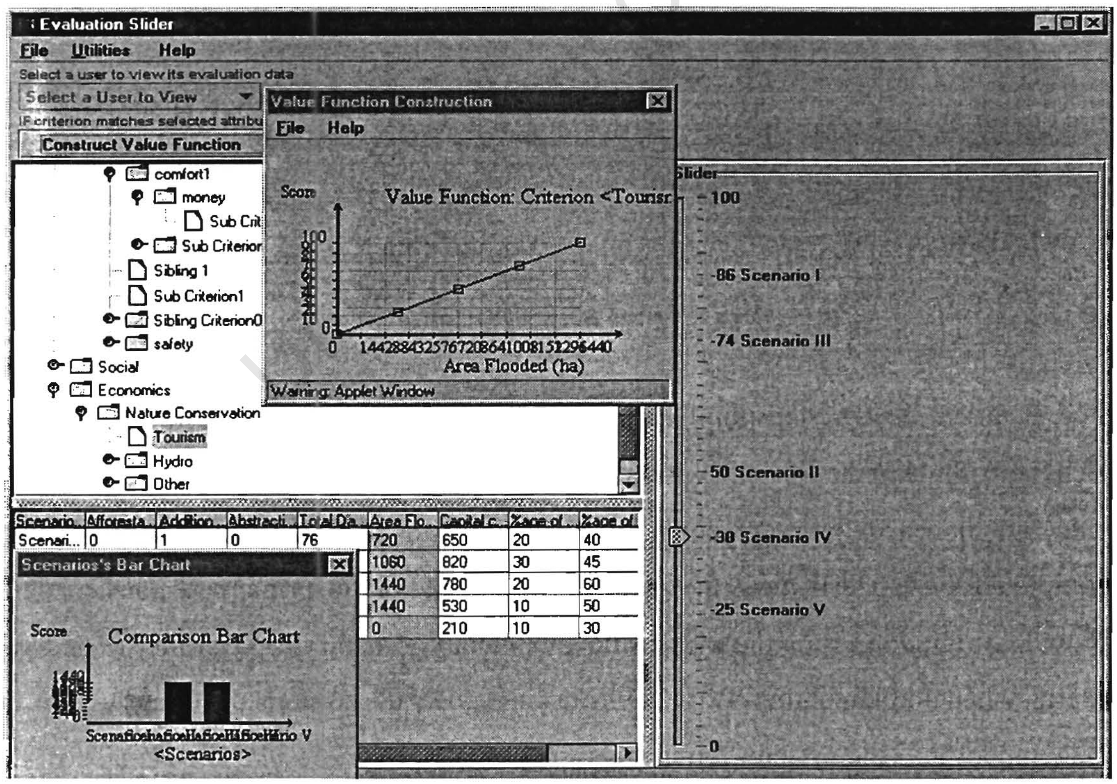


Figure 8.6: Evaluation

The system analysis and primary design of WRC DSS were carried out based on the DSS model built in the study. System requirements and relevant classes were obtained from the existing documentation of the DSS model. Specifications of system requirements were captured with use cases and case descriptions, which need to be fulfilled by classes and class interactions. Various classes and class relationships were thereafter defined based on the DSS model. The design of the system was then started. The subsystems and the system architecture of WRC DSS was designed based on the DSS model by examining the classes identified and after checking some specific technical implementation considerations.

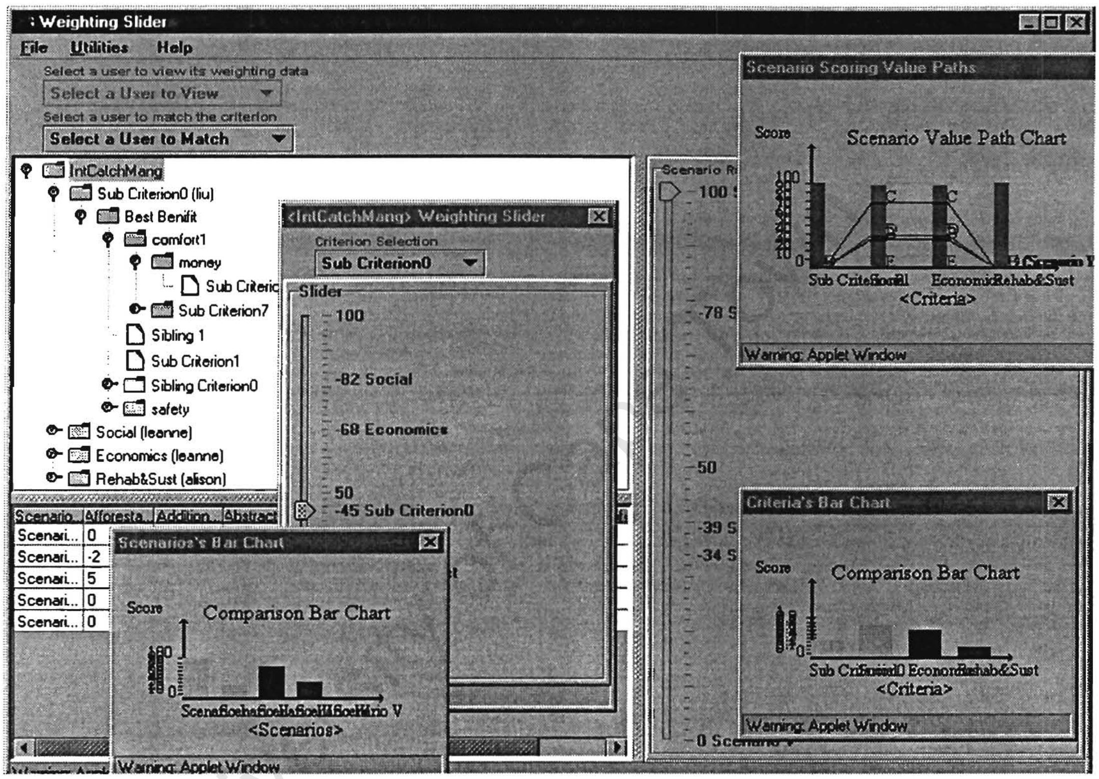


Figure 8.7: Aggregation

The detailed design and implementation of WRC DSS were facilitated by the technical implementation checklist discussed in Section 8.2.2. The technical issues listed in the checklist were considered for the system implementation so that the completed system can meet the technical DSS evaluation principles. The checklist provided a guideline for the system development based on the DSS model and was useful when implementing WRC DSS as it helped the system developers map the DSS model into the system. After the system analysis and design, the actual work of coding of the system was carried out in Java. The detailed development of WRC DSS is described in Appendix H.

8.2.5 Efficiency and Effectiveness of DSS Development

The efficient and effective way to develop a DSS is to be able to reuse existing knowledge during various phases of system analysis, design and implementation. Reuse in the analysis phase might take the form of retrieving and copying previous commercial or non-commercial analysis documentation while reuse in the design and implementation may take advantage of commercial, proprietary, or home-developed documentation as well as class libraries, which are collections of classes stored in an organised way. The DSS model built in the study is the first and essential step towards the development of reusable documentation and DSS class libraries. Generic aspects of DSS analysis and design are documented and defined in object orientation, and can then be implemented with specific programming languages. These classes are thereafter organised to form class libraries, which are then used to speed up the construction of specific DSSs, resulting in efficiency in system development.

The DSS model offers an efficient and effective way for the development of DSSs for natural resource management by reusing generic system analysis and design results. The primary system requirements and classes of a specific system can be easily obtained from the existing DSS model. The system subsystems and system architecture are also derived from the existing design results of the DSS model. Various aspects of analysis and design for a specific system, including system requirements, decision elements and other system components, system architecture, etc, are obtained by elaborating, translating, and copying the corresponding counterparts from the DSS model.

The system development of WRC DSS based on the DSS model has demonstrated both efficiency for the development process and effectiveness of the system developed. The system was developed with an effort of about four person-months. A person-month represents the work of one average developer in one month's time. The development effort is relatively little and therefore efficient for a system with a scale of about 20,000 source lines of code (Albrecht and Gaffney, 1983; Bassman, McGarry, and Pajerski, 1995). The implementation of the system was effective in meeting the decision making requirements and the required DSS performance evaluation principles. This is discussed in subsequent sections when a slightly modified version of WRC DSS is used to demonstrate the application of the object-oriented MCDM methodology to the decision problem of the removal of alien vegetation on Table Mountain in South Africa.

8.3 Facilitation of Decision Analysis

This section shows the practical application of the object-oriented approach in the facilitation of decision analysis. Three decision analysis workshops were conducted to evaluate the performances of the object-oriented decision analysis methodology and a DSS, which is the modified version of WRC DSS. Each workshop involved a student group (representing concerned but non-expert citizens) on a hypothetical decision problem, which was to consider strategies for the removal of alien vegetation from Table Mountain in Cape Town, South Africa.

8.3.1 The Decision Problem

Table Mountain, South Africa, is situated at the northern end of the Cape of Good Hope Peninsula and has been described as a unique natural wonder, owing to its beautiful appearance and rich and diverse flora growing in a setting of majestic proportions and extraordinary beauty. Table Mountain also provides countless recreational and tourism opportunities which are considered to be potential sources of income and job creation in the region.

During the 300-odd years since colonisation, the mountain has been exploited, burnt, eroded, vandalised and despoiled. For centuries since the 1560s, indigenous trees were cut for shipbuilding, furniture-making, housing, firewood, etc. Eventually in the 1880s, it became necessary to introduce trees to green the slopes of the mountain, to assist in controlling soil erosion, and also to provide much-needed timber. The afforestation is of importance in terms of aesthetic and shade values, but is damaging to the natural ecosystem. Alien vegetation has outcompeted the region's indigenous fynbos. The afforestation also has links to other factors which detrimentally affect the ecosystem of Table Mountain. For instance, alien trees help spread fires in the mountain and temperature of the fires has increased while fynbos actually needs fires to regenerate the frequency. Therefore, accidental fires have devastated indigenous vegetation and caused soil erosion and mudslides.

The problem of the eradication of alien plants has arisen due to the damage caused to many aspects of Table Mountain. There would have to be an acceptable balance between social, economic, and conservation goals in the decision making process of removing aliens. Rights of various stakeholders, including individual landowners and communities, must be respected, while considering the conservation of the environment. The hypothetical decision support workshops are based on this problem. Refer to Appendix I for more information about the decision problem.

8.3.2 The Workshops

Each workshop involved a group decision making procedure, which was divided into four phases, i.e., initial understanding, strategic analysis and further understanding, structuring, and exploring. In first phase, an initial understanding of the decision problem was gained from the analysis of the problem context and the people involved. In the second phase, further understanding came from the participants' examining the roles and values of the stakeholders and identification of decision options. In the third phase, concerns of each stakeholder were organised into tree structures. In the fourth phase, subjects were asked to express their preferences of decision options according to the specific concerns, and also to indicate the relative importance of the concerns. A ranked list of decision options was then generated. Refer to Chapter 7 for more information about these decision making processes. The program for each workshop was as follows (with approximate times for each stage):

(1) Introduction (15 minutes)

(2) Initial understanding (40 minutes)

- (i) Problem context
- (ii) People and stakeholders
- (iii) Criteria, option elements and options

(3) Further understanding (60 minutes)

- (i) Analysis of roles and values of stakeholders (30 minutes)
- (ii) Identification of criteria, option elements and options (30 minutes)

(4) Structuring (40 minutes)

(5) Exploring (40 minutes)

(i) Evaluation of options (20 minutes)

(ii) Weighting of criteria and choice (20 minutes)

Each workshop was in fact a simplified application of the methodology proposed. It was simplified in order to reduce frustration of subjects when solving the problem so that the workshop can be conducted in a time period much shorter compared to a real problem situation. Refer to Appendix I for more information about the workshop procedure.

In the first stages of each workshop, all subjects worked together in a collective way in finding out the general concerns and the people involved. At a later stage, each of them (or two of them as a group) was asked to represent a specific stakeholder in order to analyse and understand the problem in more detail, to find out their preferred options, to organise the specific stakeholder's concerns, and assess the options.

Some knowledge generalised in an object-oriented way was used in the workshops to facilitate the participants to initially identify and analyse some aspects of the decision problem, including the problem context, the people involved, criteria, elements of a decision alternative, decision alternatives, value trees, etc. The knowledge was obtained from the experiences of previous decision analysis cases of afforestation/deforestation and also from literature. This kind of knowledge was generalised and provided in order to give subjects general background information about some aspects of the problem. Appendix I contains details about this generalised knowledge. The concept of classes was not mentioned to subjects to avoid the explanation of abstract concepts. The name of category was used instead. For the purpose of simplicity, the knowledge was represented in a simplified way, which only contains primary information.

A computerised DSS was used to support the decision making procedure in the workshop. The system was based on WRC DSS (Section 8.2.4) with some slight modifications in order to cater for the new case. Subjects could access the system through the Internet at their own computers according to their own time schedule. But in order to participate in the decision making according to the above scheduled program, subjects were asked to sit together in the postgraduate computer laboratory of the Department of Statistical Sciences at the University of Cape Town.

Three workshops were conducted. The material in Appendix I was sent to the workshop participants some days prior to the workshops so that they were reasonably familiar with the workshop settings at the start. The workshop settings, including the generalised knowledge provided, the DSS used, and the decision making procedure followed, were the same for all three workshops. The participants in the workshops were students from the University of Cape Town. They were familiar, to varied extents, with the surroundings of Table Mountain, where the university is located. In the three workshops, different groups of students were chosen in order to examine how the methodology and the system could facilitate the decision analysis procedures for different people. The first workshop involved eight voluntary participants who were postgraduate students from several faculties at the university. The second workshop involved nine Honours students all from the Department of Statistical Sciences. The third workshop involved 12 conservation biology masters students from the Department of Zoology (who thus had some expert knowledge although not truly "domain experts").

Questionnaires were filled out by subjects at the end of the workshops to measure different aspects of the decision analysis methodology and the DSS. In the subsequent sections, the solutions reached at different workshops are discussed and the analysis of the questionnaires is conducted.

8.3.3 Workshop Results

The decision problem is a difficult and complicated natural resource management problem. Considerable effort and time have to be taken to obtain some insights and reach some solutions in the decision making processes. In the workshops, the students represented the general public instead of the experts. They are not required to have any special skills or knowledge with regard to decision analysis and the decision problem. However, the workshops were expected to produce some insightful solutions in a limited time period of about three and a half hours.

In the first workshop, the group identified seven representative stakeholders:

1) Recreational users: e.g., hikers, picnickers, birders, historians, etc.
2) Landowners and estate agency
3) Forestry companies
4) Local Government, including the Fire Department and the Water Management Department
5) Insurance companies
6) Conservationists and other researchers: e.g, biological, hydrologists, botanists, geologists, etc.
7) Commercial operators and tourism industry: e.g., Cable Way for the cable car

Seven decision options were identified:

1) Option 1: Clear all aliens (including Newlands and Cecilia Forests, which are primarily wooded recreational areas, even though planted with alien trees) over 20 years. Only in the lower areas, will some non-invasive (e.g., chestnuts, camphors) and some special invasive trees (e.g., large pines) be allowed to remain.
2) Option 2: Raise money to get option 1 done. Parts of the mountain to be made available to develop restaurants, hotels, etc, and promote tourism and broader regional benefits.
3) Option 3: Eliminate aliens over 50 years. Exclude commercial activities and Control Tourism. Create research opportunities and facilities to study indigenous flora.
4) Option 4: Continuously acquire new land for forestry (e.g. extension of Cecilia forest to the reservoirs). Improve technology to minimize land use and research into quick growers.
5) Option 5: The status Quo. Keep the extent of forestry of aliens as is.
6) Option 6: Over 25 years, clear invasive aliens and leave non-invasive aliens on the upper mountain. Clear invasive aliens except some areas and leave non-invasive aliens on the lower mountain.
7) Option 7: Create fire barriers by separating aliens into patches over 20 years, and remove big pines near residences.

Six participants out of eight worked on their own while the other two participants worked together as a team. Participants were randomly allocated to represent specific

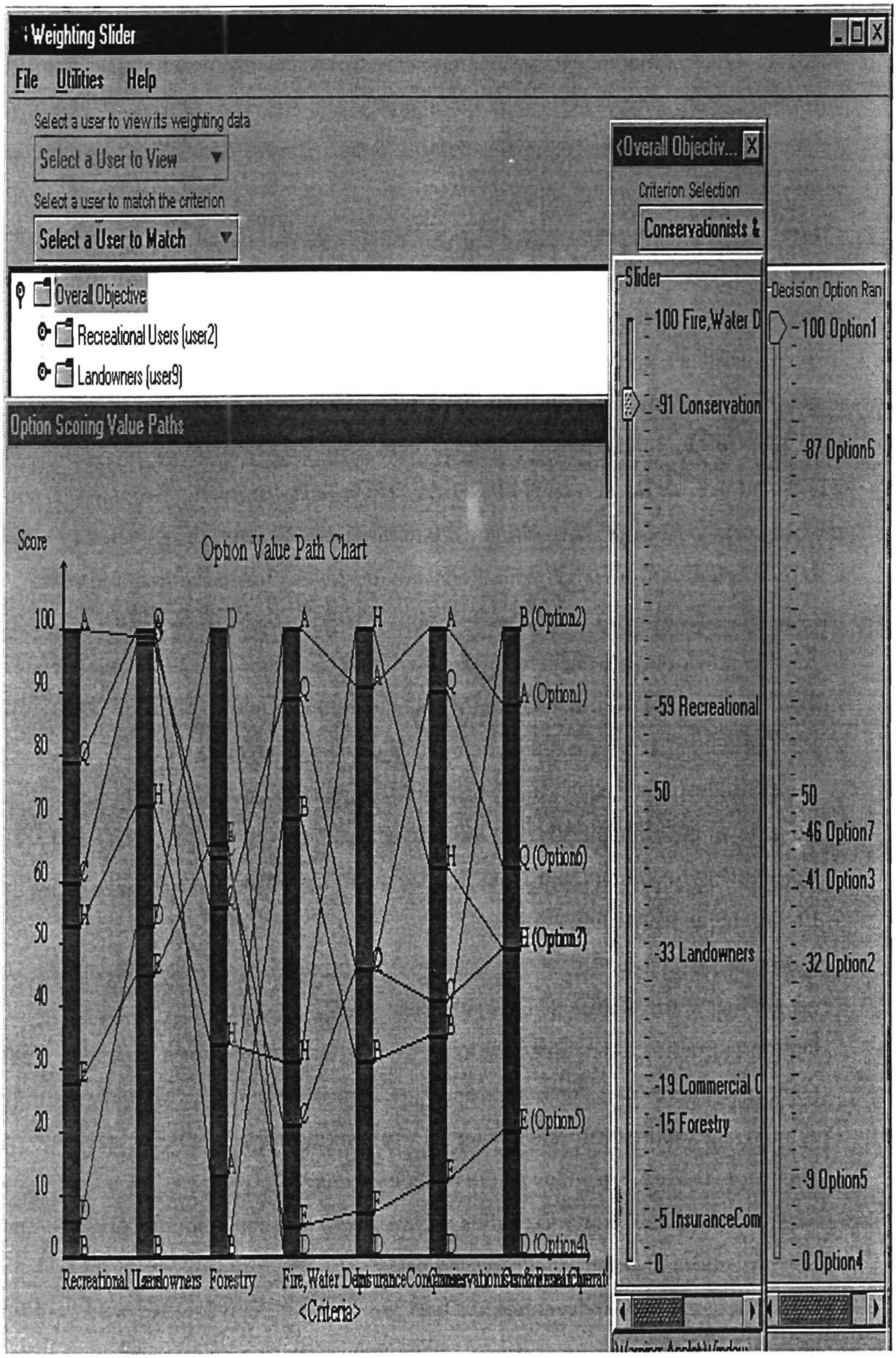


Figure 8.8: Result of the First Workshop

stakeholders. Figure 8.8 shows the overall result of the workshop. Option 1 was considered as the most preferred option. The option was close to full clearance of aliens on Table Mountain except keeping some specific aliens in some areas. Option 6 was scored as the second highest. It was about the full clearance of invasive aliens and keeping of the non-invasive aliens. Option 7 and 3 were ranked third and fourth respectively. They were for the clearance of aliens but to a less extent or over a longer period of time. Option 2 was listed above option 5 and 4. Option 2 involved commercial construction, which could be detrimental to the stakes of the primary stakeholders. Option 5 was the status quo. Option 4, which is about the extension of forestry, was least preferred. Sensitivity analysis showed that no moderate changes of relative importance of stakeholders could reshuffle the order of the options. As seen from the scoring value paths, option 2, 4 and 5 were rejected by some stakeholders while Option 6 seems acceptable to all the stakeholders. In reality, if Option 1 could not be accepted by some stakeholders, Option 6 might be accepted as a solution or as a recommendation from which variations may be created for further exploration.

In the second workshop, the participants identified six representative stakeholders:

1) Tourists, tourist agencies, and commercial operators (e.g. the cable car company)
2) Recreation, e.g. residents, hikers, mountain clerks, and cyclists
3) Timber industry, secondary industry, forestry company
4) Local level government: the Fire Department
5) Local level government: the Water Management Department
6) Environmentalists (flora, fauna)

Seven options were identified. These options were:

1) Option 1: Eradication of all aliens in 5-15 years, including all areas and along roads (especially drives used by tourists and in residential areas)
2) Option 2: Continuously clear aliens in the hiking areas but leave non-invasive aliens for shade. Keep aliens on the residential areas.
3) Option 3: Remove invasive aliens from everywhere continuously. Keep non-invasive aliens in control and assess impact.
4) Option 4: Continuously acquire new land, e.g. 20% more in some areas, in order to keep the forestry business as usual at the efficient level.

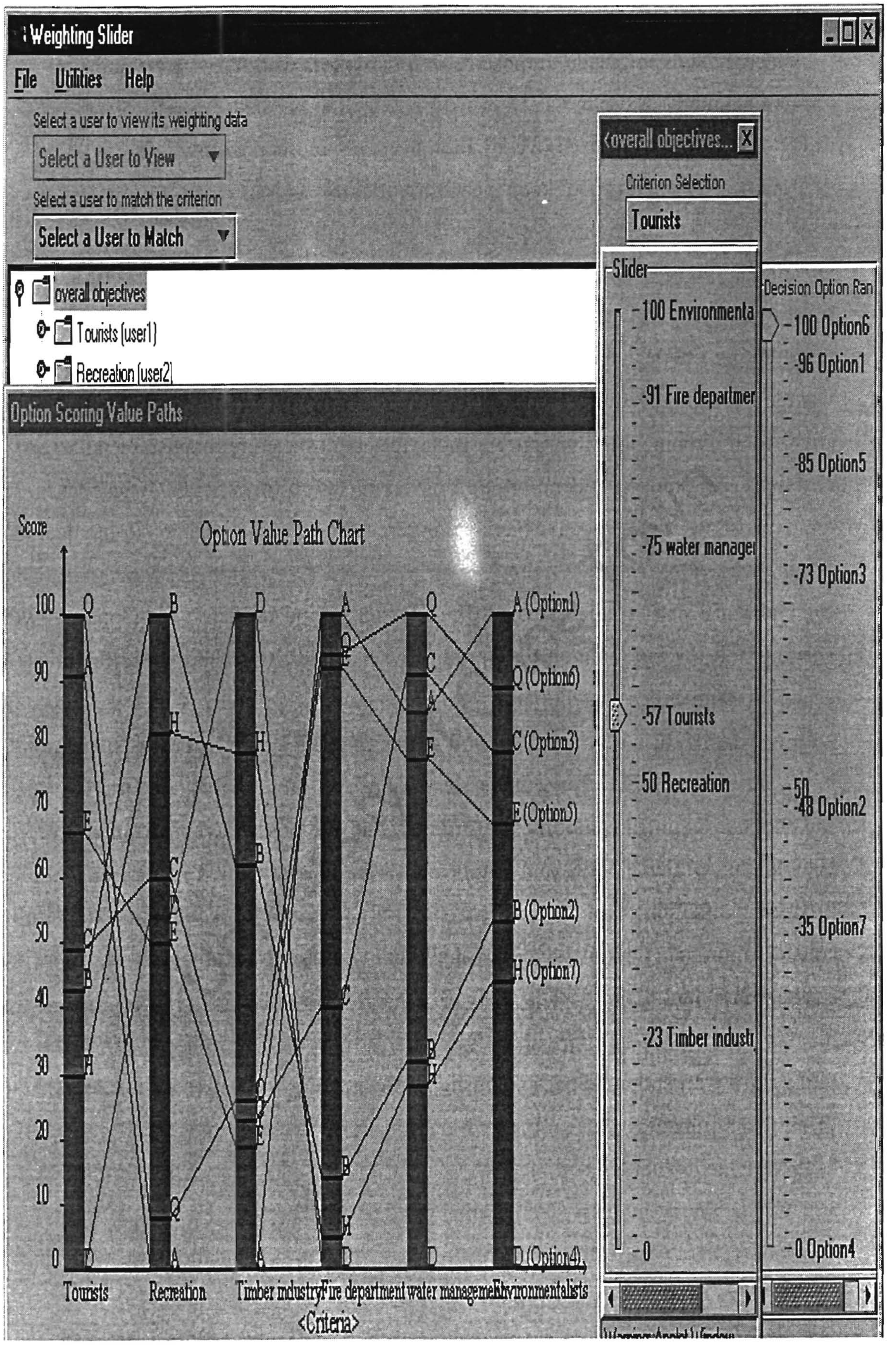


Figure 8.9: Result of the Second Workshop

5) Option 5: In 5-15 years, remove invasive aliens from everywhere. But keep non-invasive aliens for shade, impose fines for smokers who throw butts and stop urban encroachment
6) Option 6: In 25 years, remove all alien trees, up to urban edges.
7) Option 7: The status quo, keep the extent of aliens as is.

Out of the nine participants, three worked independently, each representing a single stakeholder, while the other six participants were divided into three teams, each representing one stakeholder. Figure 8.9 shows the overall analysis result. Option 6 was considered the most preferred option. This option involved eradication of all aliens up to urban edges. Option 1, which was about the comprehensive eradication of all aliens, was ranked second. Option 5 was at the third position. It was about full removal of invasive aliens while leaving non-invasive for shade. Some technical details were also added. Option 5 was followed by Option 3, which was about clearance of invasive aliens over a little longer period of time than Option 5 while keeping non-invasive in control. Option 2 was scored fifth. It proposed the aliens be cleared to a much less extent than Option 5, 1, 6 and 3. Option 7, which was the status quo, was the sixth on the rank. Option 4 was to extend the alien forests and was the least preferred option. Any moderate change of the relative importance of the stakeholders could not modify the order of the options. As seen from the scoring value paths, Option 1, 4, 6, and 7 appeared to be unacceptable to some stakeholders. In a real situation, Option 5 might probably receive more attention for further consideration than Option 1 and 6 even though Option 5 was only at the third position on the option rank.

In the third workshop, the participants identified six representative stakeholders:
1) Timber companies
2) Private landowners
3) Tourists, tourist industry, recreational users (e.g. hikers)
4) Wood sellers (e.g. selling bunches of wood on the side of the road, typically for barbecues) and harvesters of plants
5) Conservationists, academics, researchers, botany societies, birders, naturalists, etc
6) Government (local, national, and municipal) departments: the Fire Department; the Water Management Department

Seven decision options were identified:

1) Partial deforestation of commercially significant aliens in patches over the long term; plantations stay.
2) Continuously remove all invasive aliens, starting in high biodiversity zones; management and control of non-invasives in recreational and residential areas; no plantations.
3) Remove all aliens (including plantations) as soon as possible, and restore sites with indigenous vegetation.
4) Expansion and maintenance of specific zones of pine plantation with fire buffers for minimum 50 years.
5) Partial removal of invasive aliens in 15 years, i.e. remove along trails, hiking areas, etc, keep near boundaries and picnic areas; keep non-invasives.
6) Eradication of invasive aliens from most, not all, communal zones in residential areas, e.g. along drainage lines and green belts, in 15 years; cutting of fire breaks; rehabilitation immediately following clearance; keep non-invasives.
7) The status quo: keep the extent of aliens as is.

The 12 participants were divided into six teams, each including two participants and representing a stakeholder. Figure 8.10 shows the overall analysis result. Option 2 was considered the most preferred option. This option involved eradication of all invasive aliens and control of non-invasive ones. Option 6 and 5 were closely ranked second and third. They were all about partial clearance of invasive aliens, but Option 6 included establishment of fire blocks while Option 5 proposed the alien clearance to a larger extent than Option 6. Option 3 was at the fourth position, and was followed by Option 1. Option 3 was about the comprehensive eradication of all aliens. Option 1 suggested partial clearance of aliens over a long term. Option 7, which was the status quo, ranked sixth. Option 4, which was about the extension of aliens (i.e. commercial forestry), was the least preferred option. Sensitivity analysis showed that moderate changes of the relative importance of most stakeholders could not modify the order of the options except for the "private landowners" and "tourists, tourist industry, recreational users (e.g. hikers)". When "private landowners" became 50% less important, or "tourists, tourist industry, recreational users (e.g. hikers)" became 30% more important, Option 5 was preferred over Option 6 to a small extent. This was

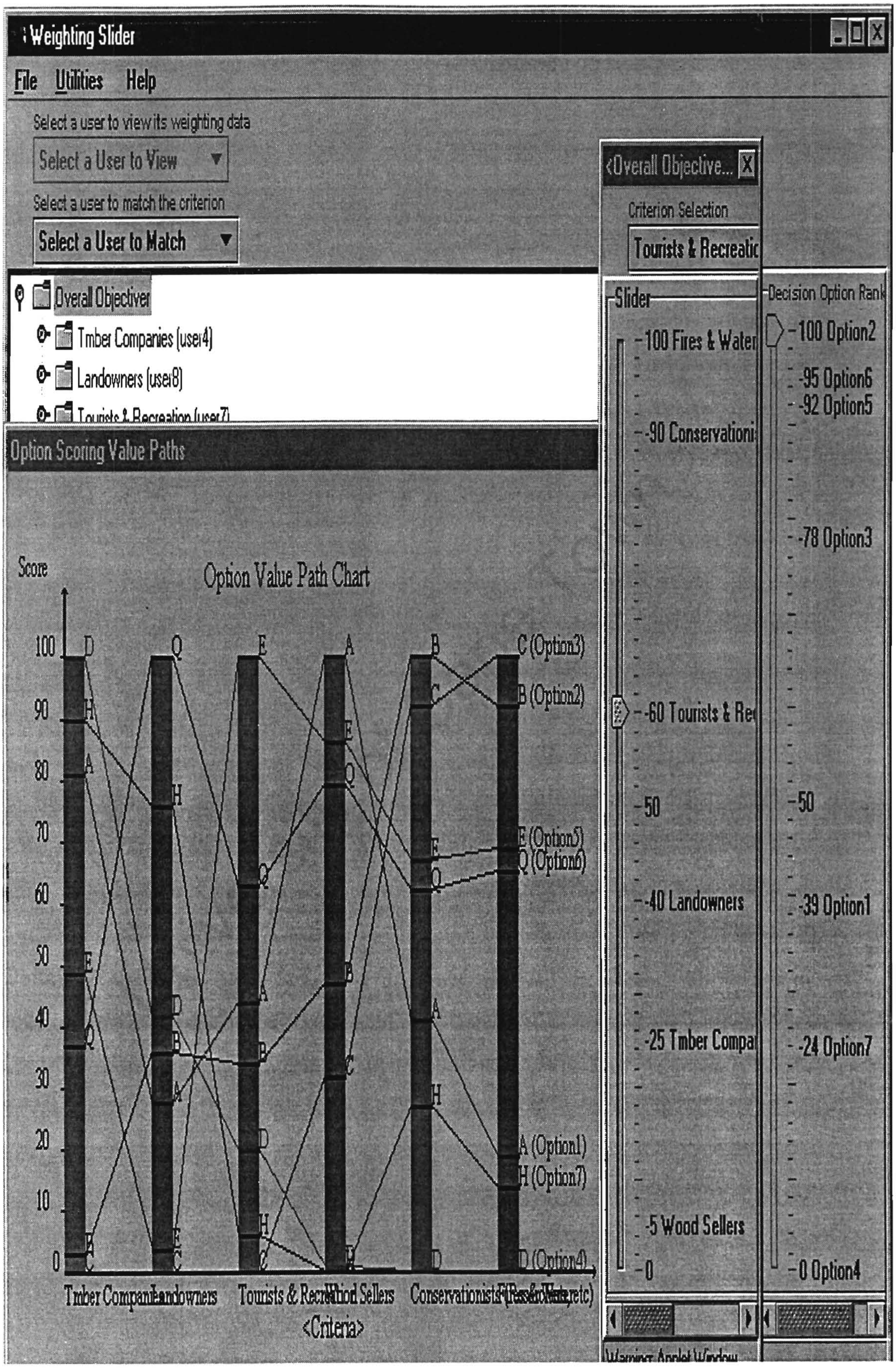


Figure 8.10: Result of the Third Workshop

because these two options, as shown on the scoring value paths, did not make much difference for most stakeholders except for these two stakeholders. The scoring value paths also showed that among the most preferred four options, Option 2, 3, and 5 appeared to be unacceptable to some stakeholders. In a real situation, Option 6 might probably receive more attention for further consideration than other options.

Two major problems arose in the workshops. Firstly, difficulties were experienced in the representation of knowledge in a formal object-oriented way. Since the participants needed to understand and reuse the knowledge in a limited period of time, the knowledge should be represented in a simple and succinct way. In fact many aspects of the knowledge were simplified in order to cater for the workshops. In a real situation, this problem would be better handled when more time is available to allow the explanation of some basic ideas of the methodology and the systematic exploration (refer to Chapter 7) of all aspects of the generalised knowledge represented in a formal object-oriented way. Secondly, it was difficult to find a solution for synchronisation in the use of a GDSS in a face-to-face workshop in which the participants worked at different paces. At some phases of decision making, some participants had to wait for the other to finish their part of work. In a real situation, this problem would be solved by allowing participants to use the system from their own computers. Participants from different regions can then access the system through the Internet at different times, carrying out their decision making tasks according to their own schedule.

Due to the different backgrounds of the participant groups, there are some differences in the outcomes of the three workshops. The first and third workshops obtained more technical details about their concerns and their options than the second one. This is probably because the participants of these two workshops had broader experiences and more knowledge of the decision context than the second group. The fact that some participants could generate more detailed decision elements does not contradict our methodology, which only offers generalised knowledge. In a real situation, most decision participants are probably well informed about their decision problems. On the other hand, the first workshop took a little longer than the second and the third workshops. One main reason was that the participants of the first workshops were less computer-skilled.

On the other hand, the outcomes of all three workshops show some consistency and insights. Firstly, the stakeholders identified in all the workshops were more or less the same. Secondly, the options identified in all three workshops were quite consistent, given the fact that the student participants were not experts and could only represent the general public in such a difficult problem, which was analysed in such a short time. This applies especially to the second group of students who had no work experience and were at a junior level compared to the others. Nevertheless, in the three workshops, different means of action to deal with the problem were taken into consideration. Thirdly, the major concerns of the removal of aliens from Table Mountain expressed by Hey (1996) were covered in all three workshops. Hey (1996) identified fires, conservation of flora and fauna, tourism, economy, recreation, and water catchment as the primary concerns. In the three workshops, the concerns of conservation and fires were given priority in terms of importance by the participants. The fact that conservation and fires were main concerns in the decision problem explains why the options identified in the workshops were ranked roughly in accordance with the clearance extent of aliens. Generally the larger the extent of clearance an option had, the higher the option was ranked, since the main concerns were in favor of clearance of aliens. The options most preferred in the workshops were those close to that of full eradication of aliens, especially invasives, but keeping limited aliens, especially non-invasives, in some areas. The options least preferred were those that proposed some extension of aliens. The option of the status quo was the second least preferred option. At the end of the workshops, participants had showed some degree of satisfaction and confidence in the solutions. This is demonstrated as a result of the analysis of the questionnaires.

To conclude, the object-oriented MCDM methodology and the DSS developed on the basis of the DSS model provided useful decision support in the workshops in the analysis of a difficult and complicated natural resource management problem, involving non-expert participants. Consistent insights were reached by the three groups of participants with different backgrounds.

8.4 Analysis of Questionnaires

A questionnaire was designed as shown in Appendix I to evaluate the performances of the object-oriented methodology and the DSS used in the workshops. It examines the personal perceptions of the workshop participants about different aspects of the methodology and the system. The perceptions were measured on a seven-point scale in which "1" means "very poor" and "7" means "excellent".

The questionnaire contains 13 questions. The primary question asked is how confident the participant feel about the solutions reached. The other 12 questions were put into two categories. The first category is about the performance of the methodology. There were four questions, including how useful the generalised knowledge provided was, how quick the understanding of the problem was achieved, how easily the problem was analysed, and how easily the decision problem was formulated. The second category is about system performance. There were eight questions with regard to the support of group decision making, the guidance in the decision making processes, the support of problem understanding, the support of brainstorming, the support of problem structuring, the support of judgement elicitation, the support of basic sensitivity analysis, and the display of the analysis result. These eight questions were based on the DSS performance evaluation items in Chapter 2. However, not all the DSS evaluation principles surveyed in Chapter 2 were covered in the questionnaire even though they provide an overview of measures for DSSs. Instead, only those that are considered essential for the demonstration of the system performance were contained in the questionnaires. The system, which was a slightly modified version of WRC DSS, was intended to demonstrate the use of the DSS model developed in the study in the implementation of a specific DSS. It was not the intention of WRC DSS to fully implement all the functions required by the surveyed DSS performance evaluation principles. Other evaluation items, such as those of information processing and user interface, are excluded in the questionnaires in order for the participants to concentrate on the evaluation of the items that are directly decision support related. Even though information processing and user interfaces are very important to DSSs, the decisive factors in the system effectiveness are the decision making support functions implemented in a system.

Questionnaires were sent to the workshop participants for evaluation. The answered questionnaires are used to assess the performances of the methodology and the system.

8.4.1 Hypotheses

The following hypotheses were developed to evaluate the methodology and the system. The first hypothesis concerns the overall confidence of the methodology and the system. The second and the third hypotheses examine the effectiveness of the methodology and the system respectively.

Hypothesis 1: The methodology and the system can produce solutions which the participant has confidence in.

This hypothesis deals with the overall confidence of the methodology and the system, and was evaluated by using the responses to the first question in the questionnaires together with Hypothesis 2 and 3.

Hypothesis 2: The performance of the methodology used for decision analysis facilitation is satisfactory, and the methodology is effective and efficient.

This hypothesis deals with the effectiveness and efficiency of the decision facilitation methodology, and was evaluated by the following sub-hypotheses, using the responses to the questions in the second category in the questionnaire.

H2.1: The generalised knowledge provided is valuable
H2.2: The decision problem can be understood quickly
H2.3: The decision problem can be analysed easily
H2.4: The decision problem can be formulated easily

Hypothesis 3: The performance of the DSS in terms of computerised decision making support is satisfactory, and the system is effective.

This hypothesis deals with the effectiveness of the system, and was evaluated by the following sub-hypotheses, using the responses to the questions in the third category in the questionnaire.

H3.1: The system effectively supports group decision making
H3.2: The system offers effective guidance in decision making processes

H3.3.1: The system provides effective facilities to help problem understanding

H3.3.2: The system effectively enables brainstorming of the problem

H3.3.3: The system furnishes effective tools for problem structuring

H3.4.1: The system offers effective judgement elicitation tools

H3.4.2: The system provides easy sensitivity analysis

H3.4.3: The system effectively presents the results of alternative selection

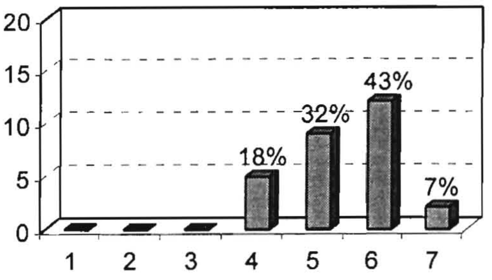
8.4.2 Results

Twenty-eight questionnaires were collected from the 29 participants of the three workshops. The responses to the questions in the questionnaires are summarised graphically in Figure 8.11. The reliability of the responses was evaluated using the Cronbach (1951) alpha test. Reliability assesses the internal consistency of the data; that is, how consistently individuals responds to questions. For the three hypotheses, the Cronbach alpha scores were 0.89, 0.76, and 0.87 respectively. A reliability score greater than 0.6 is considered acceptable (Nunnally, 1967).

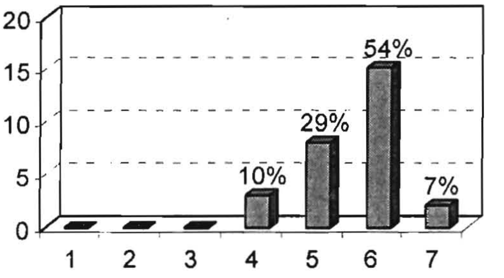
As shown in Figure 8.11, in most cases, the majority of the respondents gave a score of "6" or "7". The only substantial deviations are Question 5 and Question 11. In the case of Question 5, it is more of a recognition that the problem of formulation is intrinsically not easy. But even here the answers to Questions 2, 3 and 4 show that the respondents found the decision making methodology helpful. The responses to Question 11 were wide spread, suggesting that the tools of judgement elicitation provided in the DSS were more suitable to some respondents than the others.

It is clear that the respondents had a high level of satisfaction with regard to the various aspects of decision making stated in the questions of the questionnaires. The above hypotheses were thus supported.

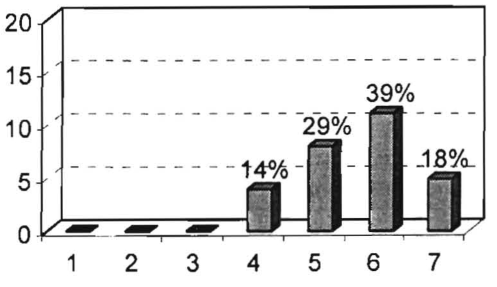
It is concluded that the methodology used for decision analysis facilitation is effective and efficient, that the DSS is effective in providing computerised decision making support, and that the methodology and the system can produce solutions which the participant has confidence in.



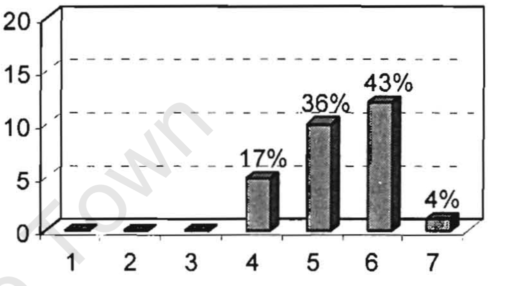
(a) Response Frequencies to Question 1



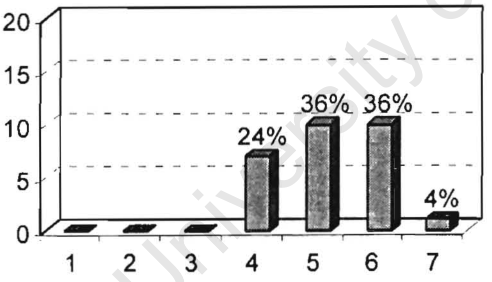
(b) Response Frequencies to Question 2



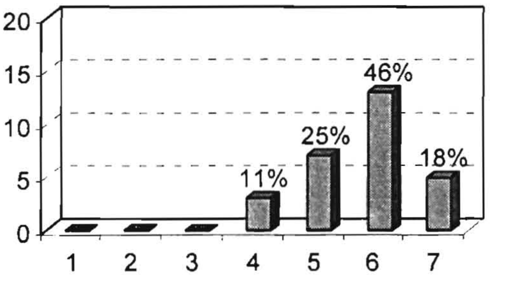
(c) Response Frequencies to Question 3



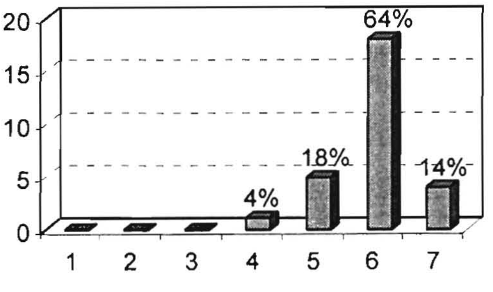
(d) Response Frequencies to Question 4



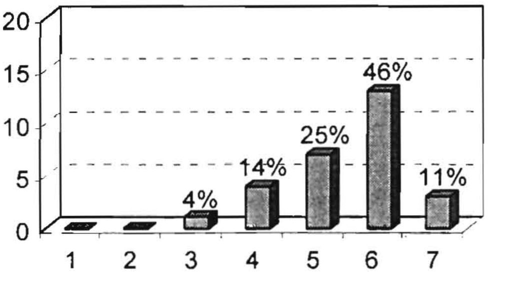
(e) Response Frequencies to Question 5



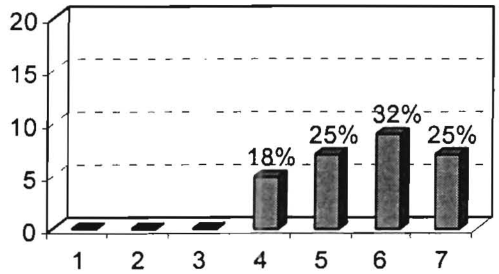
(f) Response Frequencies to Question 6



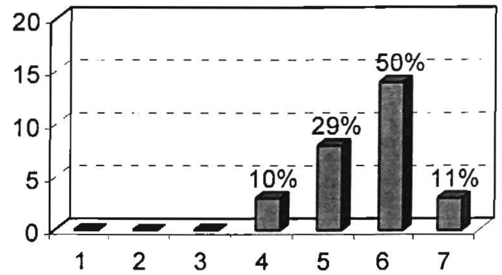
(g) Response Frequencies to Question 7



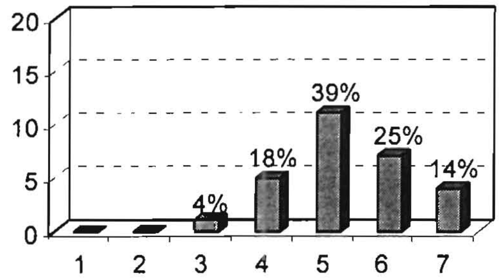
(h) Response Frequencies to Question 8



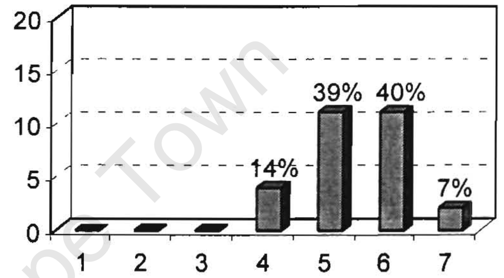
(i) Response Frequencies to Question 9



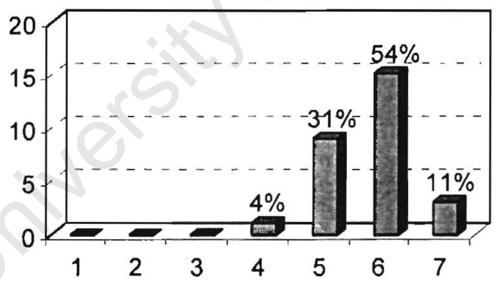
(j) Response Frequencies to Question 10



(k) Response Frequencies to Question 11



(l) Response Frequencies to Question 12



(m) Response Frequencies to Question 13

Figure 8.11: Response Frequencies to Questions 1 to 13 in the Questionnaires.

8.5 Conclusions

In this chapter, practical DSS development based on the object-oriented approach was discussed at first. The general subsystems and the system architecture were defined for DSSs for MCDM in natural resource management. They show the physical deployment and the organisation of various system components, providing a blueprint for the detailed system implementation for DSSs. A checklist of technical implementation considerations, which contain the DSS evaluation principles for technical implementation, provides a guideline in the detailed design of a specific DSS. Various aspects of analysis and design for a specific system can be obtained from the DSS model by means of elaboration, translation and copying.

The DSS model built in the study is very useful in the development of DSSs. The system analysis and design need not be invented from scratch every time such a system needs to be developed. Items of the DSS model, such as use cases, classes, class relationships, and the system architecture, are mapped from the DSS model to a real system.

A DSS was implemented to demonstrate the application of the object-oriented approach in the development of DSSs. The system was called the DSS for Water Research Commission (WRC DSS). The DSS model defined in the study played a very important role in the system design. The system requirements were quickly captured by comparing the general requirements for the DSS model and the functionality of the system. Decision elements and other system components were then identified by examining those in the DSS model, possibly with a minor modification for some components. The system architecture was obtained by checking the corresponding parts of the architecture in the DSS model. The system development was efficient due to the reuse of the generic aspects in the DSS model.

This chapter has also demonstrated the practical application of the object-oriented approach in the facilitation of decision analysis for MCDM natural resource management problems. Three decision analysis workshops were conducted by using a hypothetical decision problem. As a result of the analysis of the questionnaires completed by the participants, empirical evidence was gained about the effectiveness and efficiency of the object-oriented approach in facilitating decision analysis and also about its effectiveness in the development of the DSS, which is the slightly modified version of WRC DSS. The object-oriented approach provides a solid methodological and philosophical basis for both decision making and the development of DSSs. Conclusions of the study are contained in the next chapter.

Chapter 9

Summary, Main Conclusions and Prospects

9.1 Summary and Main Conclusions

This study has developed an approach for decision analysis and DSS modelling for MCDM in natural resource management based on object orientation, which is generally rendered as that the world and any part of it are composed of independent yet interacting objects. The main aim has been twofold: to provide a philosophical methodology for decision analysis and to find an effective and efficient way for decision analysis and DSS development for MCDM in natural resource management.

Several issues were observed as a result of the literature examination (see Chapter 2). Firstly, the undertaking of MCDM and the development of DSSs tend to be complex and inefficient, leading to low productivity in decision analysis and DSSs. Secondly, natural resource management is very complex due to its multiple dynamic aspects, and the situation of low productivity gets worse for natural resource management problems. Thirdly, there is clearly a need for philosophically sound methodologies for decision analysis and DSS development, especially for those based on MCDM natural resource management decision problems. Such a methodology should be simple, transparent, and be able to reuse the past experience and relevant knowledge. It should also uniformly model decision making and the domain of DSSs. An object orientation based methodology might be able to achieve these goals. However, little research has been reported in the literature about the comprehensive application of object orientation in decision analysis or DSS modelling.

The foundation of the methodology and the philosophy of object-oriented decision analysis and DSS analysis was first explored (see Chapter 3). It was shown that object orientation could provide a philosophy and a methodology which cater for the desirable requirements as surveyed in Chapter 2. Firstly, object orientation can naturally model the real world from different viewpoints in a simple, transparent and flexible way. Secondly, object orientation offers a mechanism to reuse the existing proven knowledge and past experiences for the understanding of decision problems, decision making and eventually the functions of DSSs. Thirdly, object orientation provides a uniform tool to model almost all the aspects of decision making and DSS development in a unified context. Besides, object orientation has many other advantages as discussed in Chapter 3.

Based on the methodology and the philosophy of object-oriented decision support, general diagrams for object-oriented problem analysis (structuring) and DSS development were presented in Chapter 3. These diagrams mainly serve as guidelines and general processes for the applications of object orientation in decision analysis, especially in problem analysis and structuring, and DSS analysis.

The object-orientated methodology was then systematically applied to problem analysis and structuring in MCDM in natural resource management in Chapter 4. It is demonstrated that the macro decision analysis system, its DSS, the decision problem, the decision context, and the entities in the decision making procedure can be represented in terms of "objects" in object orientation. It is also shown that decision problem solving, MCDM decision making procedures and decision making activities can be modelled in an object-oriented way. An overall object-oriented representation of problem analysis in natural resource management is obtained as a fundamental object-oriented model for a MCDM decision making procedure. The representation also constitutes the basis for the analysis of DSSs for natural resource management decision problems.

Classes of decision elements and primary DSS components were identified as a result of the comprehensive analysis of the generic system requirements for DSSs for MCDM natural resource management decision problems in Chapter 5. DSS evaluation principles discussed in Chapter 2 were applied to the requirement analysis to ensure that these principles are met. The general system requirements and the classes are main components of a DSS model. These classes are also reusable decision entities for MCDM in natural resource management.

The classes of decision elements and DSS components, and the relationships of these classes, play important roles in decision analysis and DSS development. Chapter 6 demonstrates an example, in which classes and their interactions can be used to determine the resources and paths for MCDM decision making in natural resource management. Besides, classes and their relationships are useful in several ways. They are a fundamental way to represent various kinds of knowledge for decision making and DSS development. Graphical representation of classes and class interactions offer an easy way to bridge the gap between decision analysts, stakeholders, domain experts, and DSS researcher. Various classes and class hierarchies provide different hierarchies and points of view for decision problems and DSSs, and offer a mechanism to manage complexity in decision analysis and DSS development. In addition, classes with significant dynamic behaviour are useful in the analysis of the progress of decision making. A general structure containing classes and their interactions was obtained for DSS modelling and MCDM in natural resource management. It was shown that the object-oriented approach of DSS modelling could be ideally integrated with that for modelling decision problems, and that the approach is an effective method required for DSS modelling to assist the development of DSSs for natural resource management.

After integrating the basic ideas proposed in the previous chapters, the methodological guidelines for object-oriented MCDM in natural resource management were described in Chapter 7 by presenting a general framework of object-oriented decision analysis. The fundamental ideas behind this framework include the object-oriented philosophy to model the world and reuse of existing classes. The framework illustrates the basic ways and general processes to utilise object orientation in decision analysis, and is able to facilitate decision making processes of problem identification, problem analysis and structuring, evaluation, choice and implementation. These decision making processes are included in the four phases of the framework, i.e., initial understanding (IU), strategic analysis - brainstorming - decision element identification (SBI), structuring, and exploring.

In the phase of initial understanding (IU), initial understanding of a decision problem may be achieved as a result of the preliminary analysis of the objects of the decision context, actors, and other decision entities. These objects are identified based on the relevant classes generated beforehand out of knowledge and past experiences.

In the phase of strategic analysis - brainstorming - decision element identification (SBI), further understanding of the decision problem under consideration is achieved by systematically considering all the objects involved after the actor-oriented object analysis, which is on the basis of various decision objects created from their classes. Brainstorming techniques are then applied to produce a complete and refined collection of decision elements for the structuring of the problem.

In the phase of structuring, a decision problem is structured out of various decision elements with the guidance of problem structuring knowledge stored in relevant classes. The tasks of structuring mainly include the construction of decision criterion hierarchies and the generation of decision alternative sets.

In the phase of exploring, a structured decision problem is explored for the profiles of various decision alternatives by various means. These means may include elicitation of preferences and judgements of decision participants, studies of various outcomes of selection by values or by actual impact on interest parties involved, implementation planning, or even partial implementation of the decision made. A solution may be recommended after observing the achievements of alternatives resulted from the exploration.

It was also shown in Chapter 7 that the object-oriented approach has many advantages in MCDM and DSS modelling, and that a uniform context could be provided for decision analysis and the development of DSSs in this framework of the object-oriented decision analysis.

Practical implementations of the object-oriented approach were demonstrated in Chapter 8. For DSS development, the general subsystems and the system architecture are defined, and a checklist of technical implementation considerations provides a guideline in the detailed design of a specific DSS. These considerations contain the DSS evaluation principles for technical implementation discussed in Chapter 2. The analysis and design for a specific DSS are obtained from the DSS model proposed in this study by means of elaboration, translation and copying.

A system called the DSS for the Water Research Commission (WRC DSS) was developed to demonstrate the application of the object-oriented approach in the development of DSSs. The system requirements, system classes, system architecture and subsystems were obtained by using the DSS model. The system development was efficient compared to other software systems with the same scale. The analysis of questionnaires by 27 users showed that the system was satisfactory.

It was shown that the DSS model built in the study is very useful in the development of DSSs. For the development of a specific DSS, its system analysis and design need not be invented from scratch. The DSS model can provide a mechanism to facilitate the development of various system items, such as use cases, classes, class relationships, subsystems, and system architecture.

Three decision workshops were conducted to demonstrate the practical application of the object-oriented approach in the facilitation of decision analysis for MCDM natural resource management problems. A simplified version of the object-oriented decision analysis methodology proposed in Chapter 7 was followed in the workshops by using a hypothetical decision problem, which was the removal of alien vegetation on Table Mountain in South Africa. The results of the workshops showed insightfulness and consistence in the solutions reached in the workshops. The questionnaires by the participants indicated that the decision facilitation method could produce quick problem understanding and easy problem analysis and formulation.

In short, empirical evidence showed the effectiveness and efficiency of the object-oriented approach in both the development of DSSs and facilitation of decision analysis, even though there were some problems in the workshops. The object-oriented approach provides a solid methodological and philosophical basis for both decision making and the development of DSSs.

9.2 Prospects

This study could be extended in several directions, some of which are outlined in this section.

The generation of decision alternatives is an extremely important part of the decision making task (Arbel and Tong, 1982; Keller and Ho, 1988). As discussed in Chapter 7, object-orientation might provide a mechanism to help alternative generation by mainly using the interactions between different decision element objects. Attributes and operations in these objects imply the existence of decision alternatives, which may be produced by the object interactions with some intelligent guidance.

The application of other strategic analysis approaches, especially soft methods (Rosenhead, 1989b), in the object-oriented decision analysis, was briefly discussed in Chapter 7. These strategic analysis approaches might be used in a subsidiary way to achieve the objective of problem understanding. For example, the technique of cognitive mapping for behavioural representation in SSM (Checkland, 1989, 1990) could diagrammatically illustrate the situation of a decision problem. Two basic ways to apply soft methods in the context of object-oriented decision analysis were proposed without much further exploration. This brings about a very interesting topic for future research.

An effective way to deal with uncertainty analysis in object-oriented decision analysis is another aspect for further research. Preliminary ideas for uncertainty analysis with object orientation were presented in Chapter 4 of the study. An uncertainty issue might be represented with outcome scenarios inside an object so as to keep options open for later resolution. With the assistance of more information and the logic reasoning of artificial intelligence, the situation will become clearer at a later stage of analysis. However, this thought needs to be fleshed out to make it operational for practical decision problem cases and to be usable in DSSs. Rough set theory (Pawlak, 1982, 1991) and fuzzy set theory (Zadeh, 1965) might contribute to representation and processing of vagueness, imprecision and uncertainty in object-oriented decision analysis.

Research output from AI, especially from expert systems, might be valuable to manipulate knowledge in object-oriented decision analysis. This is mainly due to the fact, as discussed in Chapter 3, that AI can be coherently integrated with object orientation. The techniques from AI might be able to contribute to the research of decision alternative generation, uncertainty analysis and the representation of sophisticated knowledge inside an object (see Chapter 7 for examples).

Applications of the object-oriented methodology for decision analysis and DSS modelling on other kinds of decision problems, such as manufacturing, services, medical, public policy, etc, might provide insights on the effectiveness and efficiency of the methodology. Other levels of decision making could be addressed as well, such as personal decision making, organisational decision making, and enterprise strategies. The extension of the applications of the methodology into various decision making contexts might result in the development of additional ideas about object-oriented decision analysis.

References

Ackermann, F. and Eden, C. (1994). Issues in Computer and Non-computer Supported GDSSs, *Decision Support Systems*, 12:381-390.

Ackoff, R.L. (1974). *Redesigning the Future*, Wiley, New York.

Ackoff, R.L. (1979). Restructuring the Future of Operational Research, *Journal of Operations Research Society*, 30:189-199.

Adams, D.A., Courtney, J.F. Jr. and Kasper, G.M. (1990). A Process-Oriented Method for the Evaluation of Decision Support System Generators, *Information and management*, 19:213-225.

Adams, D.A., Nelson, R.R. and Todd, P.A. (1992). Perceived Usefulness, Ease of Use, and Usage of Information Technology: A Replication, *MIS Quarterly*, 16(2):227-247.

Aiken, MW and Liu Sheng, O.R. (1991). Artificial Intelligence Based Simulation in the Design of a GDSS Idea Generation Tool, *Information and Management*, 21:279-289.

Albrecht, A.J. and Gaffney, J.E. (1983). Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation, *IEEE Transactions on Software Engineering*, 9(6):639-648.

Amarel, S. (1967). *An Approach to Heuristic Problem-Solving and Theorem Proving in the Prepositional Calculus.* In: Systems and Computer Science, Hart, J. and Tahasu, S. (eds), Toronto, University of Toronto.

Antunes, C.H., Alves, M.J., Silva, A.L. and Climaco, J.N. (1992). An Integrated MOLP Method Base Package--A Guided Tour of TOMMIX, *Computers & Operations Research*, 19(7):609-625.

Applegate, L.M., Chen, T.T., Konsynski, B.R. and Nunamaker, J.F. (1987). Knowledge Management in Organisational Planning, *Journal of Management Information Systems*, 3:20-28

Applied Decision Analysis (1996). *DPL (Decision Programming Language)*, Menlo Park, California.

Arango, G. and Prieto-Diaz, R. (1991). *Introduction and Overview: Domain Analysis Concepts and Research Direction*, pages 9-26. In: Domain Analysis and Software Systems Modelling, Prieto-Diaz, R. and Arango, G. (eds), IEEE Computer Society Press, Los Alamitos, California.

Arbel, A and Tong, RM (1982). On the Generation of Alternatives in Decision Analysis Problems, *Journal Operational Research Society*, 33:377-387.

Ariav, G. and Ginzberg, M.J. (1985). DSS Design: A Systemic View of Decision Support, *Communications of the ACM*, 28(10):1045-1052.

AT&T Bell Laboratories (1997). CADET, Murray Hill, New Jersey.

Bana e Costa, C.A. (Editor) (1990). *Reading in Multiple Criteria Decision Aid*, Springer Verlag, Berlin.

Bana e Costa, C.A., Stewart, T.J. and Vansnick, J.C. (1997). Multicriteria Decision Analysis: Some Thoughts based on the Tutorial and Discussion Sessions of the ESIGMA Meetings, *European Journal of Operational Research*, 99:28-37.

Banxia Software Ltd (1996). *Decision Explorer*, Glasgow, G4 0LT, Scotland, UK.

Barclay, S. (1987). *A User's Manual to HIVIEW*, Publ. Decision Analysis Unit, London School of Economics and Political Science, London.

Barclay, S. (1988). *A User's Manual to EQUITY*, Publ. Decision Analysis Unit, London School of Economics and Political Science, London.

Bassman, M., McGarry, F and Pajerski, R. (1995). Software Measurement Guidebook, SEL-94-102, Software Engineering Laboratory, NASA/GSFC.

Batachia, I.L. (1999). Towards User-Centred and Cost-Effective Development of Environmental Decision Support Systems, *Studies in Informatics and Control*, 8(4):279-93.

Belardo, S. and Harrald, J. (1992). A Framework for the Application of Group Decision Support Systems to the Problem of Planning for Catastrophic Events, *IEEE Transactions on Engineering Management*, 39(4):400-411.

Belton, V. (1999). *Multi-criteria Problem Structuring and Analysis in a Value Theory Framework*, pages 12-1 - 12-32. In (Gal, Stewart and Hanne, 1999).

Belton, V. and Elder, M. (1994). Decision Support Systems: Learning from Visual Interface Modelling, *Decision Support Systems*,12:355-364.

Belton, V. and Vickers, S.P. (1989). *VISA: Visual Interactive Sensitivity Analysis for Multiple Criteria Decision Aid. User Manual*, Publ. V. Belton and SPV Software Products, UK.

Benayoun, R., DE Montgolfier J., Tergny, J and Larichev, O. (1971). Linear Programming with Multiple Objective Functions: Step Method (STEM), *Mathematical Programming*, 1:366-375.

Bennett, P., Cropper, S. and Huxham, C. (1989). *Modelling Interactive Decisions: the Hypergame Focus*, pages 283-314. In (Rosenhead, 1989d).

Berztiss, A.T. (1998). Domain models for flexible decision support systems, Context Sensitive Decision Support Systems. IFIP TC8/WG8.*3 International Conference on Context-Sensitive Decision Support Systems*, Berkeley, Widmeyer, G.R, Brezillon, P. and Rajkovic, V. (eds), pages 216-26, Kluwer Academic Publishers, Dordrecht, Netherlands.

Bhargava, H.K., Sridhar, S. and Herrick, C. (1999). Beyond Spreadsheets: Tools for Building Decision Support Systems, *Computer*, 32(3):31-39.

Bidgoli, H. (1996). A New Productivity Tool for the 90's Group Support Systems, *Journal of Systems Management*, 47(4):56-62.

Biggerstaff, T.J. and Richter, C. (1987). Reusability Framework, Assessment, and Directions, *IEEE software*, 4(2):41-9.

Binbasioglu, M. (1995). Key Features for Model Building Decision Support Systems, *European Journal of Operational Research*, 82:422-437.

Blanning, R.W., Holsapple, C.W. and Whinston, A.B. (1993). Postscript, In: Special Issue on Model Management Systems, *Decision Support Systems*, 9:143-144.

Bomme, P. and Zimmermann, T. (1995). *An intelligent Symbolic Object for Decision Making Procedures*, pages 1-11. In: Developments in Computer Aided Design and Modelling for Civil Engineering, Topping, B.H.V. (ed), Civil-Comp Press, Edinburgh.

Bonczek, R.H., Holsapple, C.W. and Whinston, A.B. (1981). *Foundations of Decision Support Systems*, Academic Press, New York.

Booch, G. (1986). Object-Oriented Development, *IEEE Transaction on Software Engineering*, 12(2):211-221.

Booch, G. (1991). *Object-Oriented Design with Applications*, Benjamin Cummings, Redwood City, California.

Booch, G. (1994). *Object-Oriented Analysis and Design with Application, second edition*, The Benjamin/Cummings Publishing Company, Inc., California.

Booch, G., Rumbaugh, J., Jacobson, I. (1999). *The Unified Modelling Language User Guide*, Addison-Wesley, Massachusetts.

Borenstein, D (1998). Towards a Practical Method to Validate Decision Support Systems, *Decision Support Systems*, 23(3):227-39.

Box, G.E.P. (1980). Sampling and Bayes Inference in Scientific Modelling and Robustness, *J. Roy. Stt. Soc.*, 143:383-430.

Brill, E.D., Chang, S.Y. and Hopkins, L.D. (1982). Modelling to Generate Alternatives, *Management Science*, 28:221-235.

Buede, D. (1992). Superior Design Features of Decision Analytic Software, *Computers & Operations Research*, 19(1):43-57.

Buede, D. (1996). Decision Analysis Software Survey: Aiding Insights III, *OR/MS Today*, August, 73-79.

Buede, D. (1996). Second Overview of the MCDA Software Market, *Journal of Multi-Criteria Decision Analysis*, 5:312-316.

Buede, D. (1998). Decision Analysis Software Survey: Aiding Insights IV, *OR/MS Today*, August, 56-63.

Buede, DM (1986). Structuring Value Attributes, *Interfaces*, 16(2):52-62.

Bui, T.X. (1987). *Co-oP: A Group Decision Support System for Cooperative Multiple Criteria Group Decision Making*, Publ. Springer-Verlag (Lecture Notes in Computer Science, 290), Berlin.

Carlson, E.D. (1983). *An Approach for Designing Decision Support Systems*, pages 15-39. In: Building Decision Support Systems, Bennett, JL (ed), Addison-Wesley Publishing Company, Menlo Park, California.

Chang, A.M., Holsapple, CW and Whinston, A.B. (1993). Model Management Issues and Directions, *Decision Support Systems*, 9:19-37.

Chankong, V. and Haimes, Y.Y. (1985). *Decision Making with Multiple Objectives*, Springer-Verlag, Berlin.

Chankong, V., Haimes, Y.Y., Thadathil, J. and Zionts, S. (1985). *Multiple Criteria Optimisation: A State of the Art Review*, pages 36-90. In: Decision Making with Multiple Objectives, Chankong, V. and Haimes, Y.Y.(eds), Springer-Verlag, Berlin.

Checkland, P. (1989). *Soft Systems Methodology*, pages 71-100. In (Rosenhead, 1989d).

Checkland, P. and Scholes, J. (1990). *Soft Systems Methodology in Action*, John Wiley & Sons, Chichester.

Chen, H. (1996). An object-oriented decision support system: a case of inventory management, information technology management and organisational innovation. *Proceedings of the 1996 Information resources Management Association International Conference*, pages 308-15, Harrisburg, Idea Group.

Chen, H. and Sinha, D. (1996). An Inventory Decision Support System Using the Object-Oriented Approach, *Computers & Operations Research*, 23(2):153-170.

Churchman, C.W. (1971). *Design of Inquiring Systems*, Basic Books, New York.

Coad, P. and Yourdon, E. (1991a). *Object-Oriented Analysis*, Second Edition, Yourdon Press/Prentice-Hall, Englewood Cliffs, New Jersey.

Coad, P. and Yourdon, E. (1991b). *Object-Oriented Design*, Yourdon Press/Prentice-Hall, New Jersey.

Cohon, J.L. (1978). *Multiobjective Programming and Planning*, Academic Press, New York.

Cohon, J.L. and Marks, D.H. (1975). A Review and Evaluation of Multiobjective Programming Techniques, *Water Resources Research*, 11:208-220.

Cook, S. and Daniels, J. (1994). *Designing Object Systems: Object-Oriented Modeling with Syntropy*, Hempstead, Prentice Hall, Hemel, UK.

Corner, J.L. and Corner, P.D. (1995). Characteristics of Decisions in Decision Analysis Practice, *Journal of the Operational Research Society*, 46:304-314.

Corner, J.L. and Kirkwood, C.W. (1991). Decision Analysis Applications in the Operations Research Literature: 1970-1989, *Operations Research*, 39:206-219.

Cronbach, L.J. (1951). Coefficient Alpha and the Internal Structure of Tests, *Psychometrika*, 16(3):297-334.

Davis, F.D. (1989). Perceived Usefulness, Perceived Ease of Use, and Use Acceptance of Information Technology, *MIS Quarterly*, 13(3):319-340.

Desanctis, G. and Gallupe, B. (1987). A Foundation for Study of Group Decision Support Systems, *Management Science*, 33(5):589-609.

DeTombe, D.J. (1994). *Defining Complex Interdisciplinary Societal Problems, A Theoretical Study for Constructing a Co-operative Problem Analyzing Method: the Method COMPRAM*, Thesis Publishers, Amsterdam.

DeTombe, D.J. (1996). Compram, *A Method for Analyzing Complex Interdisciplinary Societal Problems*, pages 7-29. In: Analyzing Societal Problems, DeTombe, D.J. and van Dijkum, C. (ed), Hampp Verlag, Mering.

Dolk, D.R. and Konsynski, B.R. (1984). Knowledge Representation for Model Management Systems, *IEEE Transactions on Software Engineering*, 10(6):619-628.

Dolk, D.R. and Kotteman, J.E. (1993). Model Integration and a Theory of Models, *Decision Support Systems*, 9:51-63.

Du, T.C. (1995). *The Hybrid Decision Support System Using Neural Networks*, PhD dissertation, Arizona State University.

Dutta, A (1996). Integrating AI and Optimisation for Decision Support: A Survey, *Decision Support Systems*, 18:217-226.

Dyer, J.S. and Feinberg, A. (1972). An Interactive Approach for Multicriterion Optimisation, with an application to the Operation of an Academic Department, *Management Science*, 19:357-368.

Dyer, J.S., Fishburn, P.C., Steuer, R.E. and Wallenius, J. (1992). Multiple Criteria Decision Making, Multiattribute Utility Theory: The Next Ten Years, *Management Science*, 38(5):645-654.

Eden, C. (1986). *Problem Solving or Problem Finishing?*, In: Jackon, M and Keys, P.(eds), New Directions in Management Science, Gower, Aldershot.

Eden, C. (1989). *Using Cognitive Mapping for Strategic Options Development and Analysis (SODA)*, pages 21-42. In (Rosenhead, 1989d).

Edwards, K.A. (1995). Integrated Water Resources Development and Management, *Water Resources Management in Arid Countries, Muscat, Sultanate of Oman,* March, 321-328.

Ellis, H. (1992). Acid Rain: Operations Research Joins the Battle to Control a Serious Ecological Threat, *OR/MS Today*, August, 26-28.

Enterprise LSE Ltd (1998). *EQUITY*, Houghton Street, London WC1A 2AE, UK.

Enterprise LSE Ltd (1998). *HIVIEW*, Houghton Street, London WC1A 2AE, UK.

Eom, H.B. and Lee, S.M. (1990). A Survey of Decision Support System Applications (1971-April 1988), *Interfaces*, 20(3):65-79.

Eom, S.B., Lee, S.M., Kim, E.B. and Somarajan, C. (1998). A Survey of Decision Support System Applications (1988-1994), *Journal of the Operational Research Society*, 49(2):109-120.

Evans, G.E. and Riha, J.R. (1989). Assessing DSS Effectiveness Using Evaluation Research Methods, *Information and Management*, 16:197-206.

Expert Choice Inc (1995). *Expert Choice*, 4922 Ellsworth Ave., Pittsburgh, PA 15213, USA.

Faucheux, S. and Froger, G. (1995). Decision-Making under Environmental Uncertainty, *Ecological Economics*, 15:29-42.

Fedra, K. and Jamieson, DG (1996). An Object-Oriented Approach to Model Integration: A River Basin Information System Example, pages 669-76. In: Application of Geographic Information Systems in Hydrology and Water Resource Management, Kovar, K. and Nachtnebel, H.P. (eds), Int. assoc. Hydrologic Science, Wallingford.

Fernandez, A.A. (1996). Expert Choice, *OR/MS Today,* August, 80-83.

Finlay, P. (1994). *Introducing Decision Support Systems*, NCC Blackwell, Oxford.

Finlay, P.N. and Wilson, J.M. (1997). Validity of Decision Support Systems: towards a Validation Methodology, *Systems Research and Behavioural Science*, 14(3):169-182.

Finlay, P.N. and Wilson, J.M. (2000). A Survey of Contingency Factors Affecting the Validation of End-user Spreadsheet-based Decision Support Systems, *Journal of the Operational Research Society*, 51(8):949-58.

Fishburn, P.C. (1970). *Utility Theory for Decision Making*, Wiley, New York.

French, S. (1984). Interactive Multi-Objective Programming: its Aims, Applications and Demands, *Journal of Operational Research Society*, 35:827-834.

French, S. (1995). Uncertainty and Imprecision: Modelling and Analysis, *Journal of Operational Research Society*, 46:70-79.

French, S. (1998). Decision Analysis and Decision Support Systems, Work Paper, University of Manchester.

Friend, J. (1989). *The Strategic Choice Approach*, pages 121-158. In (Rosenhead, 1989d).

Friend, J.K. and Hickling, A. (1987). *Planning under Pressure: the Strategic Choice Approach*, Pergamon, Oxford.

Gal, T., Stewart, T.J. and Hanne, T. (editors) (1999). *Multicrteria Decision Making: Advances in MCDM models, algorithms, theory, and applications*, Kluwer Academic Publishers, Norwell, Massachusetts.

Gardner, C.L., Marsden, J.R. and Pingry, D.E. (1993). The Design and Use of Laboratory Experiments for DSS Evaluation, *Decision Support Systems*, 9(4):369-379.

Gauthier, L. and Neel, T. (1996). SAGE: An Object-Oriented Framework for the Construction of Farm Decision Support Systems, *Computers and Electronics in Agriculture*, 16(1):1-20.

Geoffrion, A.M. (1987). An Introduction to Structured Modelling, *Management Science*, 33(5):547-588.

Geoffrion, A.M. (1989). The Formal Aspects of Structured Modelling, *Operations Research*, 37(1):30-51.

Gerrity, T.P. (1971). The Design of Man-machine Decision Systems: An Application to Portfolio Management, *Sloan Management Review*, 12(2):59-75.

Ginzberg, M. and Stohr, E. (1982). *Decision Support Systems: Issues and Perspectives*. In: Decision Support Systems, Ginzberg, M., Reitman, W. and Stihr, E. (eds), North-Holland, Amsterdam.

Glover, F. and Martinson, F. (1987). Multiple-use Land Planning and Conflict Resolution by Multiple Objective Linear Programming, *European Journal of Operational Research*, 28:343-350.

Goicoechea, A., Hansen, D.R. and Duckstein, L. (1982). *Multiple Decision Analysis with Engineering and Business Applications*, John Wiley & Sons, New York.

Golden, B.L., Hevner, A. and Power, D. (1986). Decision Insight Systems for Microcomputers: A Critical Evaluation, *Computers & Operations Research*, 13: 287-300.

Gorry, G.A. and Scott Morton, M.S. (1971). A Framework for Management Information Systems, *Sloan Management Review*, 13(1):55-70.

Gossain, S. (1998). *Object Modeling and Design Strategies*, Cambridge University Press, Cambridge.

Graham, I. (1994). *Object-Oriented Methods* (Second Edition), New York, Addison-Wesley.

Greco, S., Matarazzo, B. and Slowinski, R. (1999). *The Use of Rough Sets and Fuzzy Sets in MCDM*, pages 14-1 - 14-59. In (Gal, Stewart and Hanne, 1999).

Hagmann, C. (1989). *An Object Oriented Design for Local Area Decision Network (LADN) Model for Small Group Decision Support Systems (SGDSS)*, PhD dissertation, Kansas State University.

Hallefjord, A. and Jornsten, K. (1986). A Long Range Forestry Planning Problem with Multiple Objectives, *European Journal of Operational Research*, 26:123-133.

Halsall, D.N. and Price, D.H.R. (1999). A DSS Approach to Developing Systems to Support Production Planning and Control in Smaller Companies, *International Journal of Production Research*, 37(7):1645-1660.

Hämäläinen, R.P. and Lauri, H. (1993). *HIPRE 3+ Decision Support Software vs. 3.13, User's Guide*, System Analysis Laboratory, Helsinki University of Technology.

Helsinki University of Technology (1998). Web-HIPRE 3+, System Analysis Laboratory, Helsinki University of Technology.

Hersh, M.A. (1999). Sustainable Decision Making: The role of Decision Support Systems, *IEEE Transactions of Systems, Man, and Cybernetics – Part C: Applications and Reviews*, 29(3):395-408

Hey, D (1996). *The Conservation of Table Mountain,* pages 175-184. In: The Table Mountain Book, Burman, J. (ed), Human & Rousseau, Cape Town.

Hobbs, B.F., Chankong, V., Hamadeh, W. and Stakhiv, E.Z. (1992). Does Choice of Multicriteria Method Matter? An Experiment in Water Resources Planning, *Water Resources Research*, 28:1767-1779.

Holloway, C.A. (1979). *Decision Making under Uncertainty: Models and Choices*, Prentice-Hall, Englewood Cliffs, New Jersey.

Holtzman, S. (1989). *Intelligent Decision Systems*, New York, Addison Wesley.

Howard, N. (1989). *The Manager as Politician and General: the Metagame Approach to Analyzing Cooperation and Conflict*, pages 239-262. In (Rosenhead, 1989d).

Howard, R.A. (1968). The Foundation of Decision Analysis, *IEEE Transactions on Systems Science and Cybernetics*, 4(3):1-9.

Howard, R.A. (1988). Decision Analysis: Practice and promise, *Management Science*, 34:679-695.

Howard, R.A. (1989). Knowledge Maps, *Management Science*, 35(8):903-922.

Howard, R.A. and Matheson, J.E. (1984). *Influence Diagrams*, In: Readings on the Principles and Applications of Decision Analysis, II, Howard, R.A. and Matheson, J.E. (eds), Strategic Decisions Group, Menlo Park, CA.

Huber, G. (1984). Issues in the Design of Group Decision Support Systems, *MIS Quarterly*, 8:195-204.

Huh, S.Y. (1992). Modelbase Construction with Object-Oriented Constructs, *Decision Sciences*, 24(2):409-434.

Ignizio, J.P. (1976). *Goal Programming and Extensions*, D.C. Health, Lexington, MA.

InfoHarvest Inc. (1996). *CDP (Criterium Decision Plus)*, 729 122nd Ave., NE, Bellevue, WA 98005, USA.

Jacob, V.S., Moore J.C. and Whinston, A.B. (1989). An Analysis of Human and Computer Decision-Making Capabilities, *Information and Management*, 16:247-255.

Jacobson, I. (1992). *Object-Oriented Software Engineering*. Reading, Addison-Wesley, Mass.

Janssen, R. (1991). *Multiobjective Decision Support for Environmental Problems*, PhD Dissertation, Free University of Amsterdam.

Jaumard, B., Ow, P.S. and Simeone, B. (1988). *A Selected Artificial Intelligence Bibliography for Operations Research*, In: Annals of operations research: Approaches to intelligent decision support, Hammer PL(ed), , J.C.Baltzer AG, Basel-Switzerland.

Jelassi, M.T., Jarke, M. and Stohr, E. (1985). Designing A Generalised Multiple Criteria Decision Support System, *Journal of Management Information Systems*, 1:24-43.

Keen, P. (1975). Computer-based Decision Aids: The Evaluation Problem, *Sloan Management Review*, 16(3):17-29.

Keen, P. (1981). Value Analysis: Justifying Decision Support Systems, *MIS Quarterly*, 5(1):1-16.

Keen, P. (1987). Decision Support Systems: The Next Decade, *Decision Support Systems*, 3:253-265.

Keen, P. and Scott Morton, M. S. (1978). *Decision Support Systems: An Organisational Perspective*, Addison-Wesley, Massachusetts.

Keeney, R.L. (1982). Decision Analysis: An Overview, *Operations Research*, 30:804-838.

Keeney, R.L. (1988). Structuring Objectives for Problems of Public Interest, *Operations Research*, 36:396-405.

Keeney, R.L. (1992). *Value-Focused Thinking: A Path to Creative Decisionmaking*, Harvard University Press, London.

Keeney, R.L. and Raiffa, H. (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley and Sons, New York.

Keeney, R.L., Renn, O. and von Winterfeldt, D. (1987). Structuring West Germany's Energy Objectives, *Energy Policy*, 15:352-362.

Keller, L.R. and Ho, J.L. (1988). Decision Problem Structuring: Generating Options, *IEEE Transactions on Systems, Man, and Cybernetics*, 18(5):715-28.

Kim, J.K. and Park, K.S. (1997). Neural Network-based Decision Class Analysis for Building Topological-level Influence Diagram, *International Journal of Human-Computer Studies*, 47:513-530.

Kolodner, J.L. and Simpson, R.L. (1989). The MEDIATOR: Analysis of an Early Cased-Based Problem Solver, *Cognitive Science*, 13:507-549.

Kolodner, J.L., Simpson, R.L. and Sycara-Cyranski, K. (1985). A process model of case-based reasoning in problem solving, pages 284-290, *Proceedings of IJCAI-85,* Los Angeles, California.

Korhonen, P. (1992). Multiple Criteria Decision Support: The State of Research and Future Directions, *Computers & Operations Research*, 19(7):549-551.

Korhonen., P., Moskowitz, H. and Wallenius J. (1992). Multiple Criteria Decision Support--A Review, *European Journal of Operational Research*, 63:361-375.

Kreamer, K.L. and King, J.L. (1988). Computer-based Systems for Co-operative Work and Group Decision Making, *ACM Computing Surveys*, 20(2):115-146.

Lal, H. (1992). *Object-Oriented Decision Support System in Logic Programming (for Farm Machinery)*, In: Progress in Simulation, 1, Zobrist, GW and Leonard, JV (eds), Norwood, New Jersey, Ablex.

Lamsade (1994). *ELECTRE III/IV*, Lamsade-Ura 0825, University of Paris-Dauphine, Place du Marechal Del Lattre de Tassigny, 75795, Paris Cedex 16, France.

Le Blanc, L.A. and Jelassi, M.T. (1989). DSS Software Selection: A Multiple Criteria Decision Methodology, *Information and Management*, 17:49-65.

Le Claire (1989). *Decision Support Systems: An object-Oriented Conceptual Architecture*, PhD dissertation, Oklahoma State University.

Lee, S.M. (1972). *Goal Programming for Decision Analysis*, Auerbach, Philadelphia.

Lenard, M.L. (1993). An Object-Oriented Approach to Model Management, *Decision Support Systems,* 9:67-73.

Liang, T.P. (1993). Analogical Reasoning and Case-Based Learning in Model Management Systems, *Decision Support Systems*, 10(2):137-160.

Logical Decisions (1989). *Multi-Measure Decision Analysis Software. User Manual*, Publ. Logical Decisions, 164 E. Scenic Ave., Point Richmond, CA, 94801, USA.

Lootsma, F.A., Meisner, J. and Schellemans, F. (1986). Multi-criteria decision analysis as an aid to the strategic planning of Energy R&D, *European Journal of Operational Research*, 25:216-234.

MacCrimmon, K.R. (1969). *Improving the System Design and Evaluation Process by the Use of Tradeoff Information: AN Application to Northeast Corridor Transportation Planning*, RM-5877-DOT, The Rand Corporation, Santa Monica, California.

Martin, J (1993). *Principles of Object-Oriented Analysis and Design*, Prentice-Hall, Englewood Cliffs, New Jersey.

Martin, J and O'Dell, J. (1992). *Object-Oriented Analysis and Design*, Prentice-Hall, Englewood Cliffs, New Jersey.

Martin, J and O'Dell, J. (1996). *Object-Oriented Methods: Pragmatic Considerations*, Prentice-Hall, Upper Saddle River, New Jersey.

Martin, L. (1996). *Succeeding with the Booch and OMT Methods: A Practical Approach*, Rational Software Co., Addison-Wesley, California.

Mason, R.O. and Mitroff, I.I (1981). *Challenging Strategic Planning Assumptions*, Wiley, New York.

Maxwell, D.T. (2000). Decision Analysis: Aiding Insights V, *OR/MS Today*, October, 28-35.

McGrath, J.E. (1984). *Groups: Interaction and Performance*, Prentice Hall, Englewood Cliffs, New Jersey.

Mcnamee, P. and Celona, J. (1992). *Decision Analysis with Supertree*, Strategic Decisions Group, California.

Meador, G.L. and Mezger, R.A. (1984). Selecting an End User Programming Language for DSS Development, *MIS Quarterly*, 8(4):267-281.

Mendoza, G.A. (1988). A Multiobjective Programming Framework for Integrating Timber and Wildlife Management, *Environmental Management*, 12:163-171.

Mendoza, G.A., Campbell, G.E. and Rolfe, G.L. (1986). Multiple Objective Programming: An Approach to Planning and Evaluation of Agroforestry Systems – Part 1: Model Description and Development, *Agricultural Systems*, 22:243-253.

Mendoza, G.A., Campbell, G.E. and Rolfe, G.L. (1987). Multiple Objective Programming: An Approach to Planning and Evaluation of Agroforestry Systems – Part 2: An Illustrative Example, *Agricultural Systems*, 23:1-18.

Miller, A.C., Merkhofer, M.M., Howard, R.A., Matheson, J.E. and Rice, T.R. (1976). *Development of Automated Aids for Decision Analysis*, Stanford Research Institute, Menlo Park, California.

Mintzberg, H., Raisinghani, D. and Thoret, A. (1976). The Structure of Unstructured Decision Processes, *Administrative Science Quarterly*, 21(2):246-275.

Missikoff, M. (1998). An Object-Oriented Approach to an Information and Decision Support System for Railway Traffic Control, *Engineering Applications of Artificial Intelligence*, 11(1):25-40.

Mockus, J. (1989). *Bayesian Approach to Global Optimisation*, Kluwer Academic, Dordrecht.

Moore, G.C. and Benbasat, I. (1991). The Development of an Instrument to Measure the Perceived Characteristics of Adopting an Information Technology Innovation, *Information Systems Research*, 2(3):192-222.

Moore, J.H. and Chang, M.G. (1983). *Meta-design Considerations in Building DSS*, pages 173-204. In: Building Decision Support Systems, Bennett, J.L. (ed), Addison-Wesley Publishing Company, Menlo Park, California.

Moscarola, J. (1984). *Organisational Decision Process and ORASA Intervention*, pages 169-186. In: Rethinking the Process of Operational Research and Systems Analysis, Tomlinson,R. and Kiss, I(Eds), Pergamon, Oxford.

Muhanna, W.A. (1993). An Object-Oriented Framework for Model Management and DSS Development, *Decision Support Systems*, 9:217-229.

Nunamaker, J.F., Applegate, L.M. and Konsynski, B.R. (1988). Computer-Aided Deliberation: Model Management and Group Decision Support, *Operations Research*, 36(6):826-848.

Nunnally, J. (1967). *Psychometric Theory*, McGraw Hill, New York.

O'Hagan, A. (1992). *Bayensian Numerical Analysis*, pages 345-363. In: Bayesian Statistics 4 (Bernardo, J.M., Berger, J.O., Dawid, A.P. and Smith, A.F.M. (eds), Oxford University Press, Oxford.

O'Keefe, R.M. (1989). The Evaluation of Decision-Aiding Systems: Guidelines and Methods, *Information and Management*, 17:217-226.

Palisade, Co. (1997). *DecisionTool Suite*, 31 Decker Road, Newfield, NY 14867, USA.

Pawlak, Z. (1982). Rough Sets, *International Journal of Information & Computer Sciences*, 11:341-356.

Pawlak, Z. (1991). *Rough Sets: Theoretical Aspects of Reasoning about Data*, Kluwer, Dordrecht.

Pawlak, Z. (1997). Rough Set Approach to Knowledge-based Decision Support, *European Journal of Operational Research*, 99:48-57.

Pawlak, Z. and Slowinski, R. (1994a). Rough Set Approach to Multi-attribute Decision Analysis, *European Journal of Operational Research*, 72:443-459.

Pawlak, Z. and Slowinski, R. (1994b). Decision Analysis Using Rough Sets, *International Transactions of Operational Research*, 1:107-114.

Perrings, C. (1994). *Biotic Diversity, Sustainable Development, and Natural Capital,* In: Investing in Natural Capital, Jansson, A., Hammer, M., Folke, C. and Costanza, R. (eds), Island Press, Wasgington.

Phillips, L.D. (1984). A Theory of Requisite Decision Models, *Acta Psychologica*, 56:29-48.

Phillips, L.D. (1988). *People Centred Group Decision Support,* pages 208-224. In: Knowledge Based Management Support Systems, Doukidis, G., Land, F. and Miller, G. (eds), Publ.Ellis Horwood.

Pillutla, S.N. and Nag, B.N. (1996). Object-Oriented Model Construction in Production Scheduling Decisions, *Decision Support Systems*, 18:357-375.

Pomerol, J.C. (1997). Artificial Intelligence and Human Decision Making, *European Journal of Operational Research*, 99:3-25.

Purao, S.R. (1995). *A Methodology for Distribution of Object-Oriented Applications*, PhD dissertation, The University of Wisconsin-Milwaukee.

Radcliff, B. (1986). Multi-criteria Decision Making: a Survey of Software, *Social Science Microcomputer Review,* 4:38-55.

Rafanelli, M., Ferri, F., Maceratini, R. and Sindoni,G. (1995). An Object-Oriented Decision Support System for the Planning of Health Resource Allocation, *Computer Methods and Programs in Biomedicine*, 48(1-2):163-168.

Rao, V.S. and Jarvenpaa, S.L. (1991). Computer Support of Groups: Theory-based Models for Group Decision Support Systems (GDSS), *Management Science*, 37(10):1347-1362.

Rees, J. (1985). *Natural Resources Allocation*, Economics and Policy, Methuen, London and New York.

Reimann, B.C. (1985). Decision Support for Planners: How to Pick the Right DSS Generator Software, *Managerial Planning*, 33(6):22-26.

Reimann, B.C. and Waren, A.D. (1985). User-Oriented Criteria for the Selection of DSS Software, *Communication of the ACM*, 28(2):166-179.

Riesback, K. and Schank, R.C. (1989). *Inside Case-Based Reasoning*, Lawrence Erlbaum, Hillsdale.

Rizzoli, A.E. and Young, W.J. (1997). Delivering Environmental Decision Support Systems: Software Tools and Techniques, *Environmental Modelling & Software*, 12(2-3):237-49.

Rizzoli, A.E., Davis, J.R. and Abel, D.J. (1998). Model and Data Integration and Re-use in Environmental Decision Support Systems, *Decision Support Systems*, 24(2):127-144.

Romero, C. and Rehman, T. (1987). Natural Resource Management and the Use of Multiple Criteria Decision-Making Techniques: a Review, *European Review of Agricultural Economics*, 14(1):61-89.

Rosenhead, J. (1981). Operational Research in Urban Planning, *Omega*, 9:345-64.

Rosenhead, J. (1989a). *Introduction: Old and New Paradigms of Analysis*, pages 1-20. In (Rosenhead, 1989d).

Rosenhead, J. (1989b). *Diverse Unity: the Principles and Prospects for Problem Structuring Methods*, pages 341-358. In (Rosenhead, 1989d).

Rosenhead, J. (1989c). *Robustness Analysis: Keeping Your Options Open*, pages 193-218. In (Rosenhead, 1989d).

Rosenhead, J., editor (1989d). *Rational Analysis for a Problematic World: Problem Structuring Methods for Complexity, Uncertainty and Conflict*, John Wiley& Sons, New York.

Roy, B. (1973). *How Outranking Relation Helps Multiple Criteria Decision Making*, pages 179-201. In: Multiple Criteria Decision Making, Cochrane J.L. and Zeleny M. (eds),University of South Carolina Press, Columbia, South Carolina.

Roy, B. (1990). *The Outranking Approach and the Foundations of ELECTRE Methods*, pages 155-183. In: Readings in Multiple Criteria Decision Aid, Bana e Costa C.A. (ed), Springer, Berlin.

Roy, B. (1999). *Decision-aiding Today: What Should We Expect*, pages 1-1 – 1-35. In (Gal, Stewart and Hanne, 1999).

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W. (1991). *Object-Oriented Modelling and Design*, Prentice-Hall, Englewood Cliffs, New Jersey.

Saaty, T.L. (1980). *The Analytical Hierarchy Process*, McGraw-Hill, New York.

Sandiford, F. (1986). An Analysis of Multiobjective Decision-Making for the Scottish Inshore Fishery, *Journal of Agricultural Economics*, 37:207-219.

Schank, R.C. (1982). *Dynamic Memory*, Cambridge University Press, Cambridge.

Schmoldt, D.L. and Perterson, D.L. (2000). Analytical Group Decision-Making in Natural-Resources-Methodology and Application, *Forrest Science*, 46(1):62-75.

Schneymann, A., Graves, S.C. and Murphy, F.H. (1991). Franz Edelman Award for Management Science Achievement, *Interfaces*, 21(1):1-5.

Selic, B., Gullekson, G. and Ward, P.T. (1994). *Real-Time Object-Oriented Modeling*, John Wiley & Sons, New York.

Sherman W, Paulish, D.J., Klinger M and Liao, W.P. (1995). Scenario-Driven Software Design, *Electronic Design*, 43(22):67-80.

Shlaer, S. and Mellor, S.J. (1988). *Object-Oriented Systems Analysis - Modelling the World in Data*, Englewood Cliffs, Yourdon Press, New Jersey.

Shlaer, S. and Mellor, S.J. (1992). *Object-Oriented Systems Analysis: Modelling the World in States*, Englewood Cliffs, Yourdon Press, New Jersey.

Simon, H.A. (1965). *The Shape of Automation*, Harper & Row, New York.

Simon, H.A. (1973). The Structure of Ill Structured Problems, *Artificial Intelligence*, 4:181-201.

Simon, H.A. (1977). *The New Science of Management Decision* (3rd edition), Harper and Row, New York.

Smith, G.R. (1996). *Logic Decisions for Windows,* Logic Decisions, 1014 Wood Lily Dr. Golden, CO 80401, USA.

Sprague, Jr., R.H. and Carlson, E.D. (1982). *Building Effective Decision Support Systems*, Prentice-Hall, Englewood Cliffs, New Jersey.

Sprague, R.H., Jr. (1980). A Framework for the Development of Decision Support Systems, *MIS Quarterly*, 4(4):1-26.

Sprague, R.H., Jr. and Watson, H.J. (editors) (1986). *Decision Support Systems: Putting Theory and Practice*, Prentice Hall, New Jersey.

Steuer, R.E. (1977). An Interactive Multiple Objective Linear Programming Procedure, *TIMS Studies in the Management Sciences,* 6:225-239.

Steuer, R.E. (1986). *Multiple Criteria Optimisation: Theory, Computation and Application*, John Wiley & Sons, New York.

Stewart, T.J. (1988). Experience with Prototype Multicriteria Decision Support Systems for Pelagic Fish Quota Determination, *Naval Research Logistics*, 35:719-731.

Stewart, T.J. (1992). A Critical Survey on the Status of Multiple Criteria Decision Making Theory and Practice, *OMEGA The International Journal of Management Science*, 20(5/6):569-586.

Stewart, T.J. and Brent, M. (1988). *Decision Support System for Pelagic Fish Management Policy Generation*, pages 119-129. In: Operational Research '87, Rand, G.K (Ed), Elsevier Science Publishers, Amsterdam, Netherlands

Stewart, T.J. and Joubert, A. (1999). Conflicts between Conversation Goals and Land Use for Exotic Forest Plantations in South Africa, Working Paper, University of Cape Town, Cape Town.

Stewart, T.J. and Scott, L. (1995). A Scenario-Based Framework for Multicriteria Decision Analysis in Water Resources Planning, *Water Resources Research*, 31:2835-2843.

Stewart, T.J., Joubert, A., Scott, L. and Low, T. (1996). Decision Support Procedures for Water Resources Management, Report to the Water Research Commission by the Department of Statistical Sciences, University of Cape Town, Cape Town.

Stewart, T.J., Scott L. and Iloni, K. (1993). Scenario Based Multicriteria Policy Planning for Water Management in South Africa, Report to the Water Research Commission by the Department of Statistical Sciences, University of Cape Town, WRC Report No 296/1/93, Cape Town.

Strategic Decision Group (1996). *Sensitivity/Supertree*, 2440 Sand Hill Road, Menlo Park, CA 94025, USA.

Sully, P. (1993). *Modelling the World with Objects*, Prentice Hall, New Jersey.

Sussman, P.N. (1984). Evaluating Decision Support Software, *Datamation*, 30(17):171-172.

Talcott, F.W. (1992). Environmental Agenda: The Time is Ripe for an Analytic Approach to Policy Problems, *OR/MS Today*, June, 18-24.

Taylor, J. and Taylor, W. (1987). Searching for Solutions, *PC Magazine*, 15:311-352.

Tecle, A. (1992). Selcting a Multicriterion Decision Making Technique for Watershed Resources Management, *Water Resources Bulletin*, 28:129-140.

TreeAge Software Inc (1997). *DATA (Decision Analysis by TreeAge)*, 1075 Main St., Williamstown, MA 01267, USA.

Van Gundy, A.B. (1981). *Techniques of Structured Problem Solving*, Van Nostrand-Reinhold, New York.

Van Pelt, M.J.F. (1993). Ecologically Sustainable Development Project Appraisal in Developing Countries, *Ecological Economics*, 7:19-42.

Venkatraman, S.S. (1996). A Multidimensional Framework for Group Decision Support System Research and Design, *Journal of International Information Management*, 5(1):47-59.

Ventana Corporation (1994). *GroupSystems for Windows: Quick Reference*, 1430 East Fort Lowell Rd., Tucson, Arizona 85719.

Vetschera, R. (1990). Group decision and negotiation support, *O.R. Spectrum*, 12:66-77.

Visual Thinking International Ltd (1994). *VISA (Visual Interactive Sensitivity Analysis),* 3 Robert Speck Pkwy., #900 Mississauga, Ontario L4Z 2G5, Canada.

Von Winterfeldt, D. (1980). Structuring Decision Problems for Decision Analysis, *Acta Psychologica*, 45:71-93.

Von Winterfeldt, D. and Edwards, W. (1986). *Decision Analysis and Behavioural Research*, Cambridge University Press, Cambridge.

Wang, S. (1995). Decision Support Systems Domain Analysis and Modelling: An Object-Oriented Technique, *Information and Systems Engineering*, 1:289-301.

Waren, A.D. and Reimann, B.C. (1985). Selecting DSS Generator Software: A Participative Process, *Policy and Information*, 9(2):63-76.

Watson, S.R. and Buede, D.M. (1988). *Decision Synthesis: the Principles and Practice of Decision Analysis*, Cambridge University Press, Cambridge.

Waxlax, J. (1993). An Object-Oriented DSS for Strategic Management, *Computers and Industrial Engineering*, 25(1-4):573-576.

Weiss, J.J. (1985). Gentree: A Content-Oriented aid to Decision Problem Structuring, pages 453-477. In: Applications in Artificial Intelligence, Andriole, S.J. (ed), Princeton, Petrocelli, New Jersey.

Weiss, J.J. and Kelly, C. (1980). *RSCREEN and OPGEN: Two Problem Structuring Aids Which Employ Decision Templates*, Decision and Designs, Inc. McLean, VA, TR80-4-97.

Woolley, R.N. and Pidd, M. (1981). Problem Structuring- A Literature Review, *Journal of Operational Research Society*, 32:197-206.

Zadeh, L. (1965, Fuzzy Sets, *Information and Control*, 8:338-353.

Zeleny, M. (1975). The Set of All Nondominated Solutions in the Linear Cases and a Multicriteria Simplex Method, *Journal of Mathematical Analysis and Applications*, 49:430-468.

Zionts, S. and Lotfi, V. (1989). Recent Development in Multiple Criteria Decision Making, *OriON*, 5:1-23.

Zionts, S. and Wallenius, J. (1976). An Interactive Programming Method for Solving the Multiple Criteria Problem, *Management Science*, 22:652-663.

Appendix A: Use Case and Primary Case Descriptions of DSSs for Natural Resources Management

Use Case: System Administrators - Registration of users

Case Description:

System administrator logs in to the system with a predefined password at the first time

System administrator may changes his/her own profile

System administrator registers/ modifies system users

Use Case: System Administrators - Monitoring of system status

Case Description:

System administrator monitors the status of users

System administrator checks the availability of system resources

(System administrator fixes any problems)

Use Case: Facilitator/analyst – Configuration of user privileges

Case Description:

Facilitator/analyst checks/modifies user privileges

System makes the configuration effective

Use Case: Facilitator/analyst – Send messages

Case Description:

Facilitator/analyst sends a message to a specific user

Facilitator/analyst broadcasts a message letting the users share information

Use Case: Facilitator/analyst - Retrieve messages

Case Description:

Facilitator/analyst reads the new messages

Facilitator/analyst reads previous messages/information

Use Case: Facilitator/analyst – Guide for brainstorming of the problem

Case Description:

Facilitator/analyst guides the brainstorming of criteria
Facilitator/analyst guides the brainstorming of decision attributes
Facilitator/analyst guides the brainstorming of alternatives
Facilitator/analyst guides the brainstorming of rules for the attributes, relationships among the criteria and constraints that bound alternatives
Facilitator/analyst guides the brainstorming of uncertainties, event probabilities, and uncertainty dependencies

Use Case: Facilitator/analyst – Guide for generation of alternatives
Case Description:
Facilitator/analyst guides the formulation of alternative elements
Facilitator/analyst guides the formulation of alternatives
Facilitator/analyst guides the use of the alternative generation module

Use Case: Facilitator/analyst – Guide for construction of criterion hierarchies
Case Description:
Facilitator/analyst guides the formulation of criteria
Facilitator/analyst guides the formulation of criterion hierarchies
Facilitator/analyst guides the use of the criterion hierarchy construction module

Use Case: Facilitator/analyst – Guide for uncertainty expression and management
Case Description:
Facilitator/analyst guides the expression of uncertainties by decision participants at various times
Facilitator/analyst guides the exploration of uncertainties during the course of problem analysis.

Use Case: Facilitator/analyst – Guide for expression of criterion relationships
Case Description:
Facilitator/analyst guides the expression of criterion relationships by decision participants at the brainstorming stage
Facilitator/analyst guides the exploration of criterion relationships by decision participants at the value tree construction stage

Use Case: Facilitator/analyst – Co-ordination of decision making activities

Case Description:

Facilitator/analyst checks the status of decision making activities

Facilitator/analyst informs decision participants of the status

Facilitator/analyst initiates new transactions of decision making

Use Case: Facilitator/analyst - Initialisation of relevant data

Case Description:

Facilitator/analyst inputs/modifies an initial set of criteria

Facilitator/analyst inputs/modifies an initial set of decision attributes and their ranges (thresholds)

Facilitator/analyst inputs/modifies an initial set of alternatives

Facilitator/analyst inputs/modifies an initial set of rules for the attributes, relationships among the criteria and constraints that bound the alternatives

Facilitator/analyst inputs/modifies initial uncertainties, event probabilities, and uncertainty dependencies

Use Case: Facilitator/analyst – Generation of alternatives for evaluation

Case Description:

Facilitator/analyst browses the alternatives generated by stakeholders

Facilitator/analyst creates a set of representative alternatives

Facilitator/analyst sends the set to stakeholders and domain experts for reviewing

Facilitator/analyst revises the set out of the feedback

Facilitator/analyst creates a final alternative set for evaluation

Use Case: Facilitator/analyst – Construction of the system level criterion hierarchy

Case Description:

Facilitator/analyst enters the editor for criteria hierarchies

Facilitator/analyst creates the hierarchy

Facilitator/analyst modifies the hierarchy by selecting a predefined criterion or by inputting a new criterion

Facilitator/analyst saves the hierarchy

Use Case: Facilitator/analyst – Evaluation of overall criteria and final aggregation

Case Description:

Facilitator/analyst checks/modifies the system level hierarchy of criteria
Facilitator/analyst checks the progress of the evaluation for each stakeholder
Facilitator/analyst checks the relationships between criteria
Facilitator/analyst obtains the weights for the system criteria
Facilitator/analyst seeks census about the weights from stakeholders
Facilitator/analyst calculates the weighted achievement for each alternative
Facilitator/analyst ranks the alternatives in textual and graphical documentation in one of the three formats: choice, sorted alternatives, and ranked alternatives (the evaluation result for the system)

Use Case: Facilitator/analyst – Sensitivity analysis

Case Description:

Facilitator/analyst checks the progress of the final aggregation
Facilitator/analyst examines the sensitivity of the weighted achievement of alternatives to key variables

Use Case: Facilitator/analyst – Decision model input

Case Description:

Facilitator/analyst enters the editor for decision models
Facilitator/analyst inputs data and elements of a model
Facilitator/analyst saves the model

Use Case: Facilitator/analyst – decision making guidance

Case Description:

Various decision making processes update their status in the progress trace machine
Facilitator/analyst checks for the decision making status.
System automatically informs the progress status and other information of decision making
System automatically offers guidance for decision making

Use Case: Facilitator/analyst – Problem orientation

Case Description:

Facilitator/analyst check for the problem information.

Case documents show the related context, documents and data in textual and graphical ways

Use Case: Facilitator/analyst –Previous case demonstration

Case Description:

Facilitator/analyst asks for the demonstration of a previous problem case

Case documents retrieve the case base

Case documents demonstrate relevant information about the problem case

Use Case: Facilitator/analyst – Analysis result report

Case Description:

Facilitator/analyst asks for the report of the decision analysis result

System presents the textual and graphical documentation of the decision analysis result

Use Case: Stakeholder – Send messages

Case Description:

Stakeholder sends a message to a specific user

Stakeholder broadcasts a message letting other users share information

Use Case: Stakeholder - Retrieve messages

Case Description:

Stakeholder reads the new messages

Stakeholder reads previous messages/information

Use Case: Stakeholder - Brainstorming of the problem

Case Description:

Stakeholder inputs/modifies a set of criteria

Stakeholder inputs/modifies a set of decision attributes and their ranges under the guidance of the facilitator/analyst

Stakeholder inputs/modifies a set of alternatives

Stakeholder inputs/modifies a set of rules for the attributes, relationships among the criteria and constraints that bound alternatives under the guidance of the facilitator/analyst

Stakeholder inputs/modifies uncertainties, event probabilities, and uncertainty dependencies under the guidance of the facilitator/analyst

Use Case: Stakeholder – Construction of a criterion hierarchy

Case Description:

Stakeholder enters the criteria hierarchy editor

Stakeholder creates or opens the hierarchy

Stakeholder modifies the hierarchy by selecting a predefined criterion or by inputting a new criterion

Stakeholder saves the hierarchy

Use Case: Stakeholder – Generation or modification of alternatives

Case Description: (Generate a new set)

Stakeholder enters the editor for the alternative set

Stakeholder creates a new set

Stakeholder adds an alternative to the set with references to the value thresholds for the attributes

Stakeholder modifies the set

Stakeholder saves the set

Case Description: (Add extra alternatives to the set defined by the facilitator/analyst)

Stakeholder enters the editor for the alternative set

Stakeholder adds alternatives without modification to the system set and with references to the value thresholds for the attributes.

Stakeholder notifies the facilitator/analyst of the addition

Stakeholder saves the set

Use Case: Stakeholder – Evaluation of alternatives

Case Description:

Progress trace machine checks the progress of the construction of the criterion hierarchy and the generation of the alternative set.

Stakeholder indicates his/her own judgements/preferences to the alternatives in the system alternative set according to each criterion

Stakeholder saves the evaluation result of the alternatives

Use Case: Stakeholder – Evaluation of criteria and individual aggregation

Case Description:

Stakeholder checks/modifies the hierarchy of criteria

Progress trace machine checks the progress of the construction of the criterion hierarchy, the generation of the alternative set, and the evaluation for the stakeholder

Stakeholder inputs the weights for the criteria

Stakeholder calculates the weighted achievement for each alternative

Stakeholder ranks the alternatives in one of the three formats: choice, sorted alternatives, and ranked alternatives

Stakeholder saves the aggregation result (the evaluation result)

Use Case: Stakeholder – Sensitivity analysis

Case Description:

Progress trace machine checks the progress of the aggregation

Stakeholder examines the sensitivity of the weighted achievement of alternatives to key variables

Use Case: Stakeholder – Uncertainty expression and management

Case Description: (Express uncertainty)

Stakeholder enters the editor for uncertainties

Stakeholder inputs uncertainties for models, data, and judgements

Stakeholder saves the uncertainties

Case Description: (take uncertainty into consideration during various processes)

Stakeholder takes the uncertainties into consideration when generating an alternative set

Stakeholder takes the uncertainties into consideration when constructing a criterion hierarchy

Stakeholder takes the uncertainties into consideration when evaluating the alternatives

Stakeholder takes the uncertainties into consideration when evaluating the criteria

Use Case: Stakeholder – Expression of criterion relationships

Case Description:

Stakeholder enters the editor for criterion relationships

Stakeholder indicates the relationships between pairs of criteria.

Stakeholder saves the relationships.

Use Case: Stakeholder – Decision making guidance

Case Description:

Various decision making processes update their status in the progress trace file

Stakeholder checks for the decision making status.

System automatically informs the progress status and other information of decision making

System automatically offers guidance for decision making

Use Case: Stakeholder – Problem orientation

Case Description:

Stakeholder checks for the problem information.

Case documents show the related context, documents and data in textual and graphical ways

Use Case: Stakeholder –Previous case demonstration

Case Description:

Stakeholder asks for the demonstration of a previous problem case

Case document retrieve the case base

Case document demonstrate relevant information about the problem case

Use Case: Stakeholder - Analysis result report

Case Description:

Stakeholder asks for the report of the decision analysis result

System presents the textual and graphical documentation of the decision analysis result

Use Case: Domain Expert – Send messages

Case Description:

Domain Expert sends a message to a specific user

Domain Expert broadcasts a message letting other users share information

Use Case: Domain Expert - Retrieve messages

Case Description:

Domain Expert reads the new messages

Domain Expert reads previous messages/information

Use Case: Domain Expert - Brainstorming of the problem

Case Description:

Domain Expert inputs/modifies a set of criteria

Domain Expert inputs/modifies a set of decision attributes and their ranges (thresholds) under the guidance of the facilitator/analyst

Domain Expert inputs/modifies a set of alternatives

Domain Expert inputs/modifies a set of rules for the attributes, relationships among the criteria and constraints that bound alternatives under the guidance of the facilitator/analyst

Domain Expert inputs/modifies uncertainties, event probabilities, and uncertainty dependencies under the guidance of the facilitator/analyst

Use Case: Domain Expert – Construction of a criterion hierarchy

Case Description:

Domain Expert enters the criteria hierarchy editor

Domain Expert creates or opens the hierarchy

Domain Expert modifies the hierarchy by selecting a predefined criterion or by inputting a new criterion

Domain Expert saves the hierarchy

Use Case: Domain Expert – Generation or modification of a set of alternatives

Case Description: (Generate a new set)

Domain Expert enters the editor for the alternative set

Domain Expert creates a new set

Domain Expert adds an alternative to the set with references to the value thresholds for the attributes

Domain Expert modifies the set

Domain Expert saves the set

Case Description: (Add extra alternatives to the set defined)

Domain Expert enters the editor for the alternative set

Domain Expert adds alternatives without modification.

Domain Expert saves the set

Use Case: Domain Expert – Evaluation of alternatives

Case Description:

Domain Expert indicates his/her own judgements/preferences to the alternatives according to each criterion

Domain Expert saves the evaluation result of the alternatives

Use Case: Domain Expert – Evaluation of criteria and individual aggregation

Case Description:

Domain Expert checks/modifies the hierarchy of criteria

Domain Expert inputs the weights for the criteria

Domain Expert calculates the weighted achievement for each alternative

Domain Expert ranks the alternatives in one of the three formats: choice, sorted alternatives, and ranked alternatives

Domain Expert saves the aggregation result (the evaluation result)

Use Case: Domain Expert – Sensitivity analysis

Case Description:

Progress trace machine checks the progress of the aggregation

Domain Expert examines the sensitivity of the weighted achievement of alternatives to key variables

Use Case: Domain Expert – Check of the decision elements brainstormed

Case Description:

Domain Expert enters the editor for decision elements (decision attributes, criteria, etc)

Domain Expert checks decision elements with domain knowledge

Domain Expert informs relevant decision participants of expert opinions

Use Case: Domain Expert – Check of the alternatives generated and the alternative set constructed

Case Description:

Domain Expert enters the editor for alternatives

Domain Expert checks the alternatives with domain knowledge

Domain Expert checks the alternative set with domain knowledge

Domain Expert informs relevant decision participants of expert opinions

Use Case: Domain Expert – Check of the overall and individual interest objective (criterion) hierarchies

Case Description:

Domain Expert enters the editor for criterion hierarchies

Domain Expert checks the hierarchies with domain knowledge

Domain Expert informs relevant decision participants of expert opinions

Use Case: Domain Expert – Uncertainty expression and management

Case Description: (Express uncertainty)

Domain Expert enters the editor for uncertainties

Domain Expert inputs uncertainties for models, data, and judgements

Domain Expert saves the uncertainties

Case Description: (take uncertainty into consideration during various processes)

Domain Expert takes the uncertainties into consideration when generating an alternative set

Domain Expert takes the uncertainties into consideration when constructing a criterion hierarchy

Domain Expert takes the uncertainties into consideration when evaluating the alternatives

Domain Expert takes the uncertainties into consideration when evaluating the criteria

Use Case: Domain Expert – Expression of criterion relationships

Case Description:

Domain Expert enters the editor for criterion relationships

Domain Expert indicates the relationships between pairs of criteria.

Domain Expert saves the relationships.

Use Case: Domain Expert – Value threshold setting

Case Description:

Domain Expert enters the editor for the decision attributes

Domain Expert inputs value thresholds for action elements (alternative attributes)

Domain Expert saves the value thresholds.

Use Case: Domain Expert – Decision model input

Case Description:

Domain Expert enters the editor for decision models

Domain Expert inputs data and elements of a model

Domain Expert saves the model

Use Case: Domain Expert – Decision making guidance

Case Description:

Various decision making processes update their status in the progress trace

Domain expert checks for the decision making status.

System automatically informs the progress status and other information of decision making

System automatically offers guidance for decision making

Use Case: Domain Expert – Problem orientation

Case Description:

Domain expert checks for the problem information.

Case documents show the related context, documents and data in textual and graphical ways

Use Case: Domain Expert –Previous case demonstration

Case Description:

Domain expert asks for the demonstration of a previous problem case

Case documents retrieve the case base

Case documents demonstrate relevant information about the problem case

Use Case: Domain Expert - Analysis result report

Case Description:

Domain Expert asks for the report of the decision analysis result

System presents the textual and graphical documentation of the decision analysis result

Use Case: Implementation Agent – Send messages

Case Description:

Implementation Agent sends a message to a specific user

Implementation Agent broadcasts a message letting other users share information

Use Case: Implementation Agent - Retrieve messages

Case Description:

Implementation Agent reads the new messages

Implementation Agent reads previous messages/information

Use Case: Implementation Agent – Decision implementation

Case Description:

Implementation Agent checks the information and data of during all the phases of the decision making

Implementation Agent checks the final selection result.

Implementation Agent documents a plan for the implementation

Use Case: Implementation Agent - Decision analysis report

Case Description:

Implementation Agent asks for the report of the decision analysis result

System presents the textual and graphical documentation of the decision analysis result

Use Case: Observer – Send messages

Case Description:

Observer sends a message to a specific user

Observer broadcasts a message letting other users share information

Use Case: Observer - Retrieve messages

Case Description:

Observer reads the new messages

Observer reads previous messages/information

Use Case: Observer – Trial use of the system as a facilitator/analyst, a stakeholder, or a domain expert

Case Description:

Observer uses all functions of a facilitator/analyst, a stakeholder, or a domain expert in a simulation way.

Use Case: External System – Exportation of data to the DSS

Case Description:

Stakeholders and facilitator/analysts import data or files from other systems at various phases during the decision making process

Use Case: External System – Importation of data from the DSS

Case Description:

Stakeholders, implementation agent and facilitator/analysts export data or files to other systems at various phases

Appendix B: Classes of Decision Elements and Primary DSS Components for MCDM in Natural Resource Management

Appendix B-1: Decision element classes of decision problems and DSSs

AchievementMeasure: An algorithm to measure the relative degrees to which alternatives satisfy the aspirations or desires of a stakeholder, representing his/her overall strengths of preference between outcomes.

ActionRule: Constraints among different action elements.

AlternativeConstraint: Constraints that bound the alternatives.

AttributeValueThreshold: Value ranges which action elements fall in.

Choice: The final chosen alternative.

Criterion: a principle allowing comparison of decision alternatives. It is a tool to compare alternatives according to a particular significance point of view.

CriterionHierarchy: The hierarchy of organised relevant criteria.

CriterionRelationship: Relationships between criteria, such as neutral, destructive and constructive. Stakeholders can use the relationships to structure the value hierarchy or identify conditional probabilities in a useful way.

DecisionAlternative: which is called Alternative previously in Chapter 4. A description of one possible plan of action for the future.

DecisionAlternativeSet: A set of alternatives for individual users. It is also called Alternative Set.

DecisionAttribute: which is called Action Element previously in Chapter 4. It indicates an aspect of the action elements whose collective set constitutes the vector of attributes for the decision alternative. Decision attributes are the features or properties used to describe a decision alternative, for the purposes of identifying the most desirable decision alternative. It might be cardinal (a direct numerical measure), ordinal (a rank ordering), or nominal (unordered classes).

DecisionModel: Automated algorithms whereby data can be analysed in response to the evaluation of alternatives and criteria. Models may be retrieved and maintained by the model management in DSSs.

DecisionParticipant: A virtual parent class for the classes that participate the decision making.

DomainExpert: A person who had the expertise of the problem domain.

EvaluationData: The evaluation data of alternatives and their corresponding criteria at different stages.

EvaluationResult: The evaluation outcomes of various stakeholders and the facilitator/analyst. It might in one of the three forms: choice, ranked alternatives, and sorted alternatives. It corresponds to class Choice defined in Chapter 4.

EventProbability: The probability of a certain event that might affect the consequences of alternatives.

Facilitator/analyst: A person who facilitates the decision making processes.

Implementation: A set of schemes to plan the implementation of the decision choice made.

ImplementationAgent: A person or organisation who implements the decision made.

InterestAlternativeSet: A set of alternatives generated by individual users. All alternatives created by different interest groups in a system have to be considered before a final version of system level alternative set is constructed.

InterestCriterion: Criteria of the individual stakeholder or domain expert's level, which usually reflects different concerns of individuals or groups.

InterestCriterionHierarchy: The criterion hierarchy of each interest group

InterestEvaluationResult: The evaluation outcomes for individual decision participants. It corresponds to class Interest Choice defined in Chapter 4.

Judgement: Stakeholders' general judgement on alternatives or criteria.

Preference: Stakeholders' preferences to the alternatives according a certain criterion.

RankedAlternativeList: A ranked list of alternatives according to a set of criteria.

SortedAlternativeSet: A set of sorted categories of alternatives according to a set of criteria.

Stakeholder: A real world person who is able to express preferences of alternatives and usually has a stake in the problem concerned.

SystemAlternativeSet: An overall representative set of alternatives of the system. The final version of this set is used by all users to evaluate the alternatives in it.

SystemCriterion: Criteria of the system level, which usually summarises different interests of individuals or groups.

SystemCriterionHierarchy: The hierarchy of organised relevant system criteria. It is called Overall Criterion Hierarchy previously in Chapter 4, and it represents the overall hierarchy of criteria that are associated with different interest groups.

SystemEvaluationResult: The overall evaluation outcomes for the system. It corresponds to class Overall Choice defined in Chapter 4.

Uncertainty: Uncertain data or information that needs to be clarified to evaluate the alternatives. It might be due to imprecision of judgements about preferences, value and other subjective belief, lack of information and the randomness of processes.

UncertaintyDependency: A special relationship amongst uncertainties and between a certain uncertainty and other information. It specifies that a change in one thing (uncertainty or other data) may affect another thing, but not necessarily the reverse.

ValueModel: A function to measure achievement of decision alternatives according to specific models.

Weight: An indicator of importance of a criterion in comparison with others in the same level of a criterion hierarchy.

Appendix B-2: Other primary system component classes of DSSs

CaseDocument: A library of evaluation data and result and other relevant information of various problem cases.

CurrentCase: The decision problem facing the current decision participants and its related data. Relevant information and data about the problem is included.

Data: Data that keep the information for the system, such as alternative data and evaluation data. It may be exported to and imported from external systems.

ExternalSystem: External software systems such as MicroSoft Excel and Word.

File: Files that keep data in the system. They may be exported to and imported from external systems.

Message: Message sent and retrieved by individual system users.

Observer: A person who wants to study the system for various reasons.

PastCase: A problem case similar to the one under consideration. It may be resulted from previous studies by the same decision participants or someone else, and may contain case information, evaluation data, and evaluation result.

ProblemCase: A decision problem in natural resource management domain. It may include the relevant information to describe a problem case.

ProgressTrace: A library to keep the knowledge of the decision making procedure and the status of the current decision making activities. It also contains the stages into which the whole decision making procedure is divided, and the instructions given to users as to how to do next in the decision making processes.

Report: A textual and/or graphical report of the decision analysis result, including various data and information produced during the decision making procedure.

SystemAdministrator: A person who administrates the system

SystemStatus: The running status of the system.

User: A virtual parent class for all system user classes.

UserConfigure: The configuration of access and communication priorities for users such as stakeholders, facilitators, and domain experts etc.

Appendix C: Class Attributes and Operations of the DSS Model

AchievementMeasure

Attributes: name, stakeholderName, alternative, value, algorithm, description

Operations: input(), modify(), check(), calculate(), isComplete()

ActionRule

Attributes: name, sourceDecisionAttribute, affectedAttributes, preconditions

Operations: identify(), modify(), check(), examine(), delete(), save()

AlternativeConstraint

Attributes: name, validAlternativeConditions, invalidConditions

Operations: identify(), modify(), check(), examine(), delete(), save()

AttributeValueThreshold

Attributes: name, attributeName, valueProperty, valueRange

Operations: identify(), modify(), check(), examine(), delete(), save()

CriterionRelationship

Attributes: name, sourceCriterion, desCriterion, sourceToDesRelation, desToSourceRelation

Operations: identify(), modify(), check(),examine(), delete(), save()

CaseDocument

Attributes: problemCase, pastCases

Operations: retrieve(), input()

Choice

Attributes: name, stakeholderName, preferredAlternative

Operations: check(), save()

Criterion

Attributes: name, inputUser, inputTime, parentCriterion, childCriteria, description

Operations: identify(), modify(), check(), delete(), save()

CriterionHierarchy

Attributes: name, userName, criterionHierarchy, criterionRelationships

Operations: create(), modify(), check(), isComplete(), evaluate(), save()

CurrentCase

Attributes: name, information, alternativeAttributes, decisionParticipants, systemAlternativeSet, criterionHierarchy, systemCriterionHierarchy, evaluationData, evaluationResult, systemEvaluationResult

Operations: register(), modify(), orient(), delete(), check()

Data

Attributes: name, format, type, values

Operations: copy(), paste()

DecisionAlternative:

Attributes: name, decisionAttributes, attributeValues, parentAlternative, childAlternatives, alternativeConstraints

Operations: identify(), modify(), check(), delete(), save()

DecisionAlternativeSet

Attributes: name, count, alternatives

Operations: create(), modify(), evaluate(), check(), isComplete(), add(), delete(), save()

DecisionAttribute

Attributes: name, valueProperty, valueThreshold, actionRules

Operations: identify(), modify(), check(), delete(), save()

DecisionModel

Attributes: name, inputData, outputData, algorithm

Operations: identify(), modify(), calculate(), check(), delete(), save()

DecisionParticipant

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note, problemCase, userConfigure

Operations: login(), register(), modify(), checkConfigure(), modifyConfigure()

DomainExpert

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note, problemCase, userConfigure, isImitate

Operations: login(), register(), modify(), delete(), check(), checkConfigure(), modifyConfigure(), imitate()

EvaluationData

Attributes: name, stakeholderName, alternativeSet, criterionHierarchy, judgements, preferences, achievementMeasure

Operations: check(), calculate(), input(), save()

EvaluationResult

Attributes: name, stakeholderName, choice, rankedAlternativeList, sortedAlternativeSet

Operations: aggregate(), check(), save()

EventProbability

Attributes: name, event, realizationConditions, description, probability

Operations: identify(), modify(), check(), delete(), save()

Facilitator/analyst

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note, problemCase, userConfigure, isImitate

Operations: login(), register(), modify(), delete(), check(), checkConfigure(), modifyConfigure(), imitate()

File

Attributes: name, format, type

Operations: export(), import()

ImplementationAgent

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note, problemCase, userConfigure

Operations: login(), register(), modify(), delete(), check(), checkConfigure(), modifyConfigure(), plan()

InterestAlternativeSet

Attributes: name, count, alternatives

Operations: create(), modify(), check(), isComplete(), add(), delete(), save()

InterestCriterion

Attributes: name, inputUser, inputTime, parentCriterion, childCriteria, description

Operations: identify(), modify(), check(), delete(), save()

InterestCriterionHierarchy

Attributes: name, userName, criterionHierarchy, criterionRelationships

Operations: create(), modify(), check(), isComplete(), evaluate(), save()

InterestEvaluationResult

Attributes: name, stakeholderName, choice, rankedAlternativeList, sortedAlternativeSet

Operations: aggregate(), check(), save()

Judgement

Attributes: name, judge, judgeItem, referItem, valueProperty, valueRange, value, description

Operations: elicit(), modify(), check(), save()

Message

Attributes: name, priority, content, sender, receiver, sentTime

Operations: send(), broadcast(), retrieve()

Observer

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note, problemCase, userConfigure

Operations: login(), register(), modify(), delete(), check(), checkConfigure(), modifyConfigure()

PastCase

Attributes: name, information, alternativeAttributes, decisionParticipants, systemAlternativeSet, criterionHierarchy, systemCriterionHierarchy, evaluationData, evaluationResult, systemEvaluationResult

Operations: register(), modify(), orient(), demonstrate(), delete(), check()

Preference

Attributes: name, judge, judgeItem(alternative), referItem(criterion), valueProperty (preferenceProperty), valueRange(preferenceRange), value, description

Operations: elicit(), modify(), check(), save()

ProblemCase

Attributes: name, information, alternativeAttributes, decisionParticipants, systemAlternativeSet, criterionHierarchy, systemCriterionHierarchy, evaluationData, evaluationResult, systemEvaluationResult

Operations: register(), modify(), orient(), delete(), check()

ProgressTrace

Attributes: decisionStages, decisionParticipants, decisionInstructions, decisionStatuses

Operations: monitor(), instruct(), update()

RankedAlternativeList

Attributes: name, stakeholderName, rankedAlternatives

Operations: check(), save()

Report

Attributes: name, content, content

Operations: generate(), view()

SortedAlternativeSet

Attributes: name, stakeholderName, alternativeCategories, categoryValues

Operations: check(), save()

Stakeholder

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note, problemCase, userConfigure, isImitate

Operations: login(), register(), modify(), delete(), check(), checkConfigure(), modifyConfigure(), imitate()

SystemAdministrator

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note

Operations: login(), register(), modify(),delete(), check()

SystemAlternativeSet

Attributes: name, count, alternatives

Operations: create(), modify(), evaluate(), check(), isComplete(), add(), delete(), save()

SystemCriterion

Attributes: name, inputUser, inputTime, parentCriterion, childCriteria, description

Operations: identify(), modify(), check(), delete(), save()

SystemCriterionHierarchy

Attributes: name, userName, criterionHierarchy, criterionRelationships

Operations: create(), modify(), check(), isComplete(), evaluate(), save()

SystemEvaluationResult

Attributes: name, stakeholderName, choice, rankedAlternativeList, sortedAlternativeSet

Operations: aggregate(), check(), save()

SystemStatus

Attributes: userStatus, resourceStatus

Operations: monitor()

Uncertainty

Attributes: name, event, causeType, uncertaintyDependency, description, probability

Operations: identify(), modify(), check(), examine(), delete(), save()

UnceitaintyDependency

Attributes: name, uncertaintyName, sources, sourceRealizationConditions, sourceDescriptions

Operations: identify(), modify(), check(), examine(), delete(), save()

User

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note

Operations: login(), register(), modify()

UserConfigure

Attributes: accessPriority, communicationPriority

Operations: setAccessPriority(), setCommunicationPriority(), getAccessPriority(), getCommunicationPriority()

ValueModel

Attributes: name, inputData, outputData, algorithm

Operations: identify(), modify(), calculate(), check(), delete(), save()

Weight

Attributes: name, criterionName, criterionHierarchyName

Operations: input(), modify(), check(), save()

Appendix D: Class Interaction Diagrams of Primary Classes in the DSS Model

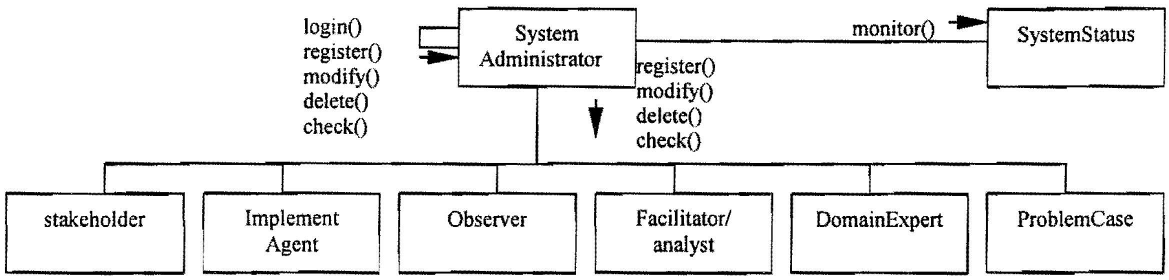


Figure D.1: Class Interaction Diagram for System Administrator

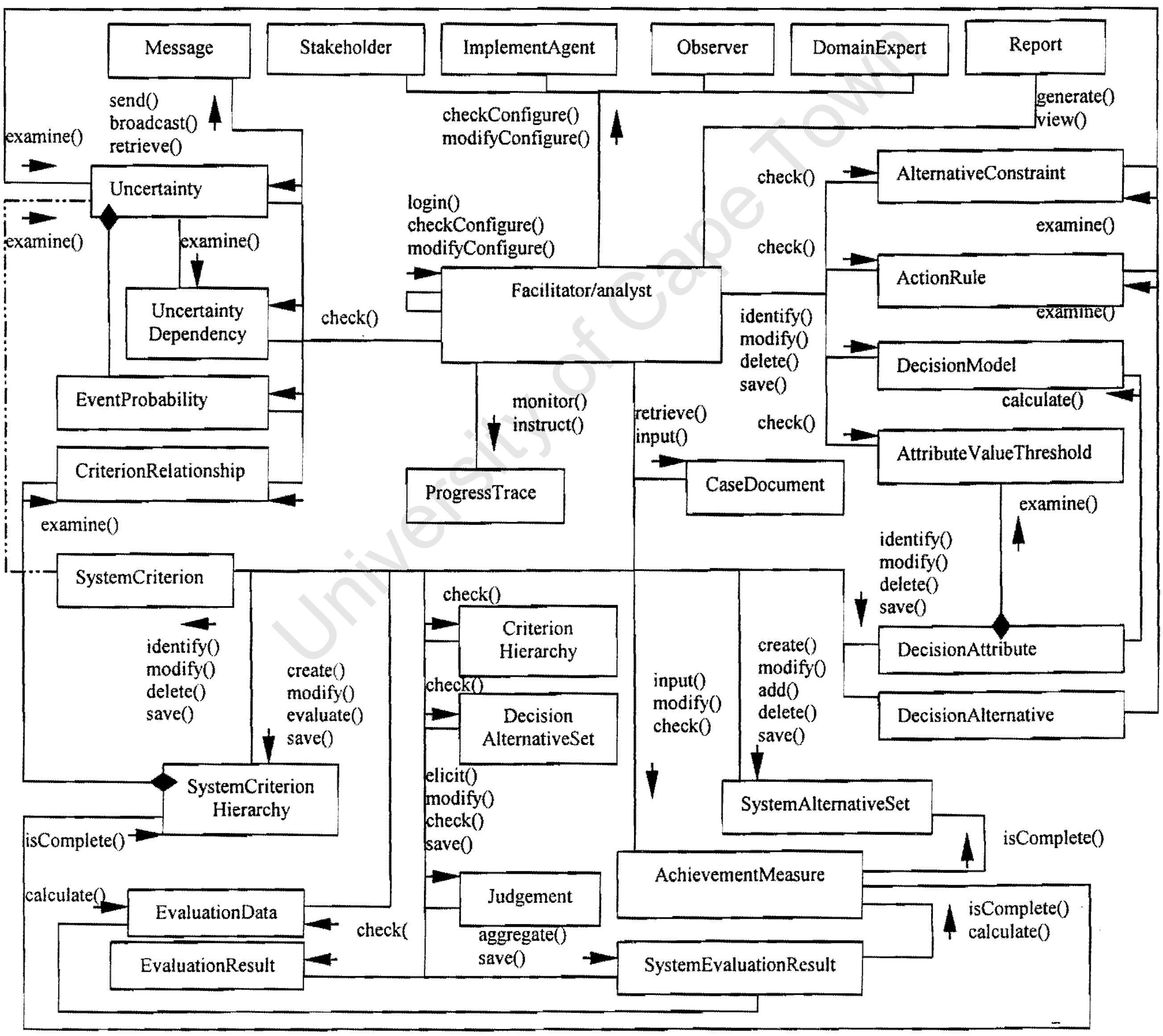


Figure D.2: Class Interaction Diagram for Facilitator/analyst

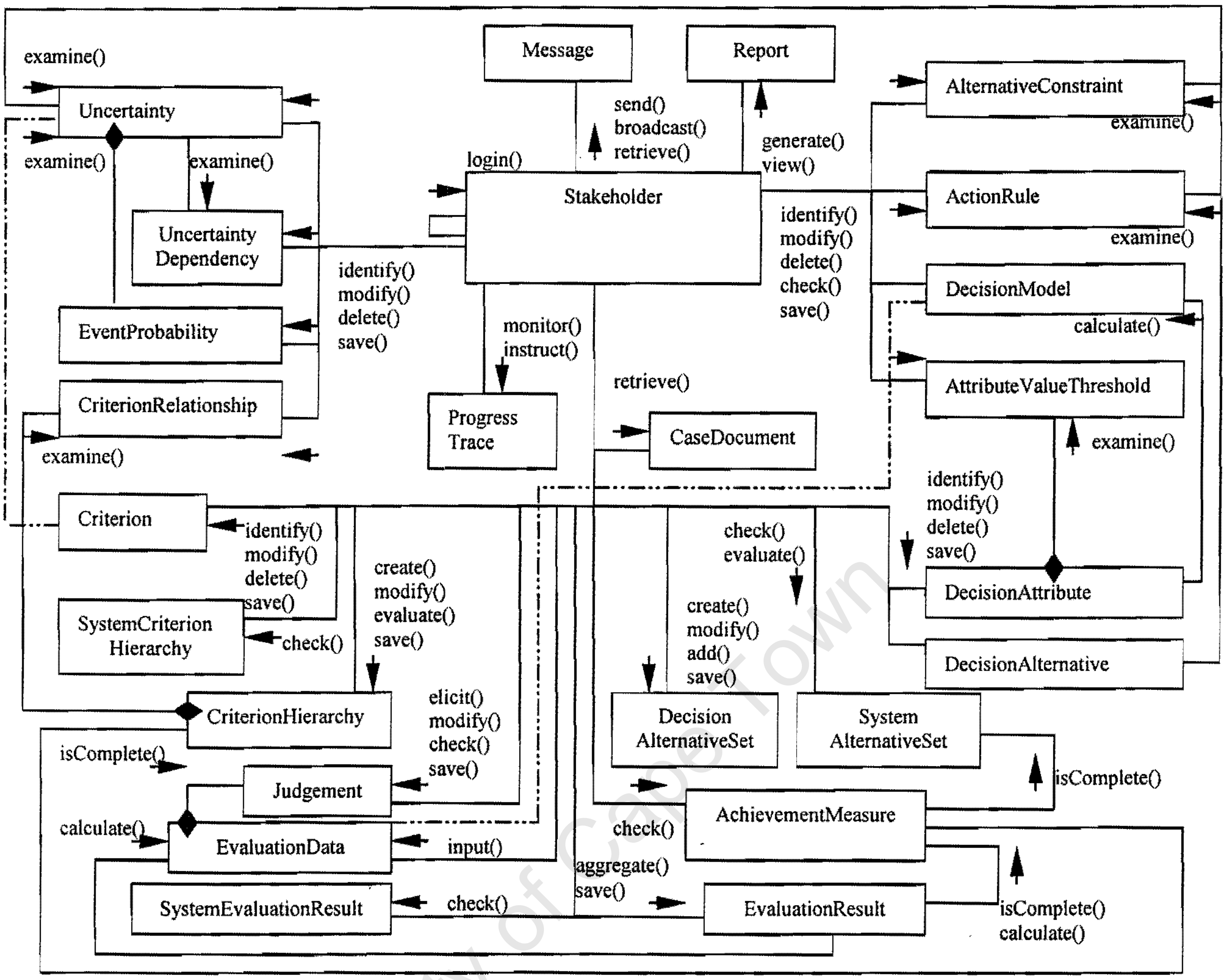


Figure D.3: Class Interaction Diagram for Stakeholder

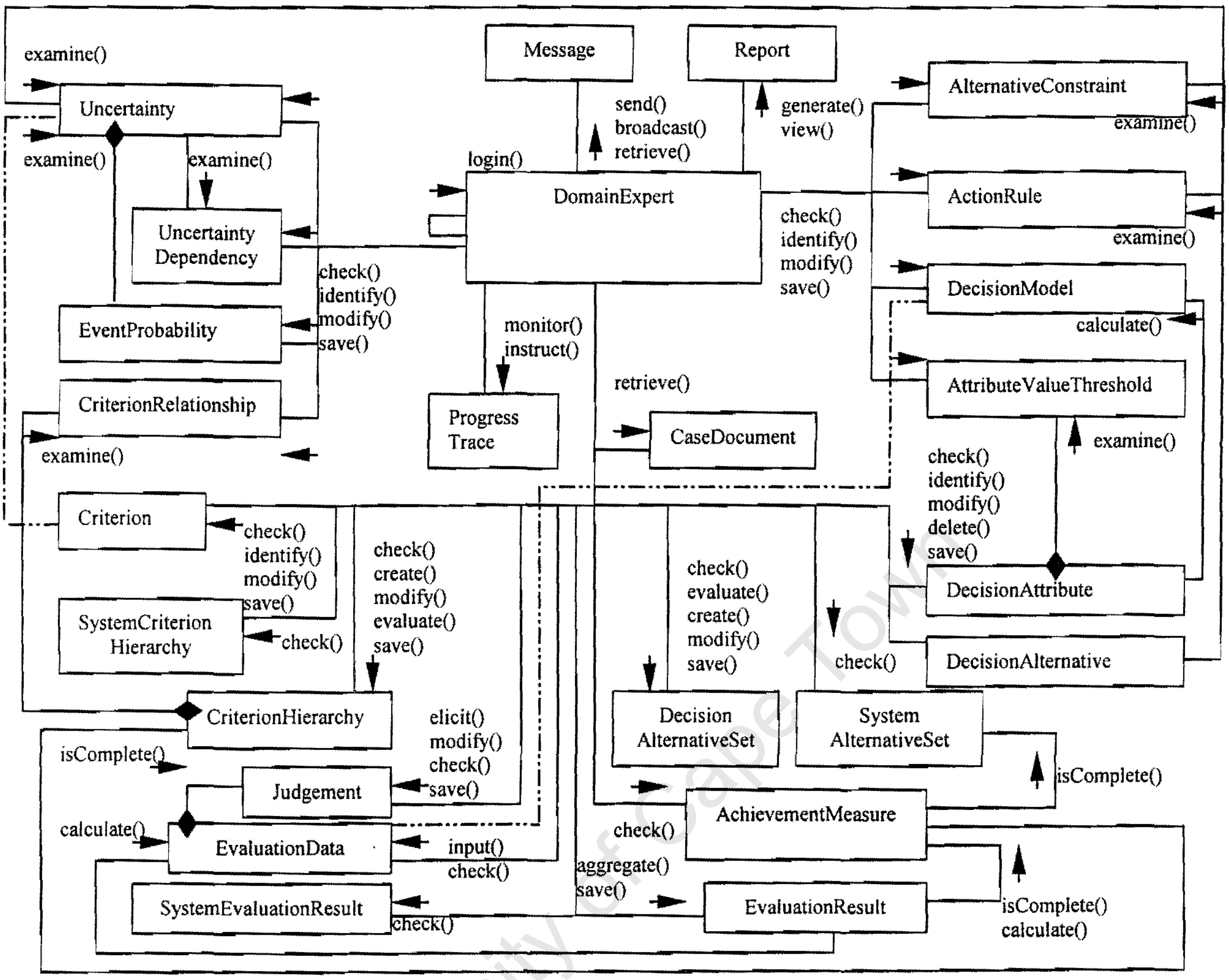


Figure D.4: Class Interaction Diagram for Domain Expert

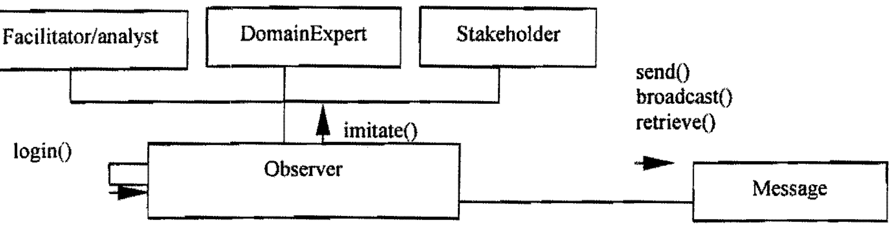


Figure D.5: Class Interaction Diagram for Observer

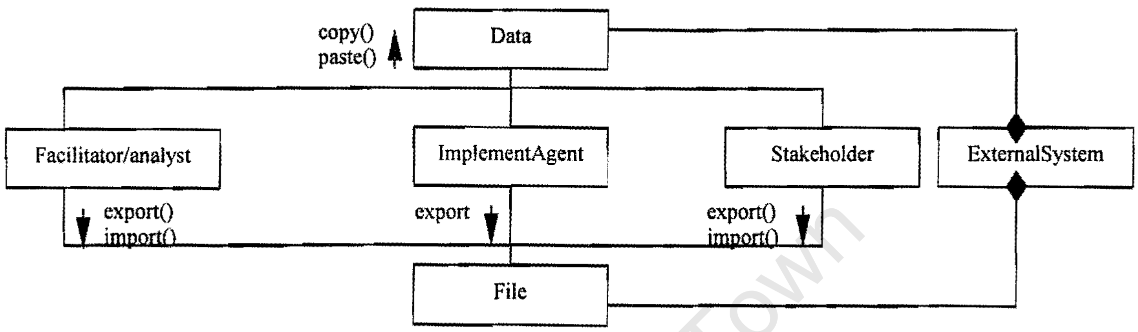


Figure D.6: Class Interaction Diagram for External System

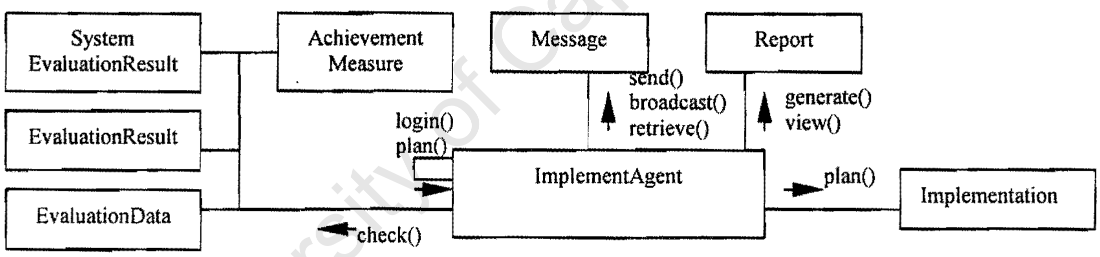


Figure D.7: Class Interaction Diagram for Implement Agent

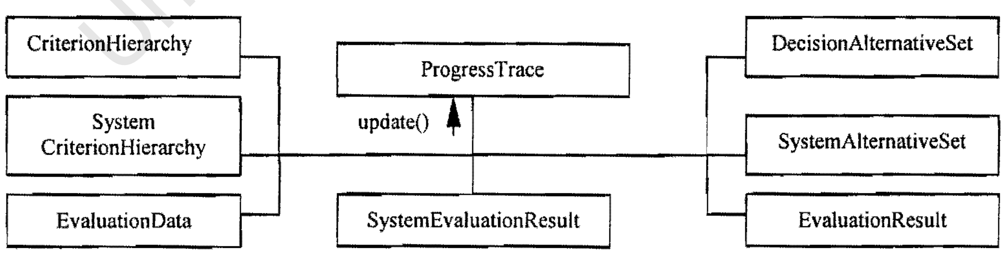


Figure D.8: Sub-diagram of Class Interactions for Progress Trace

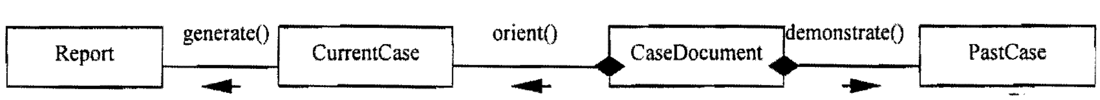
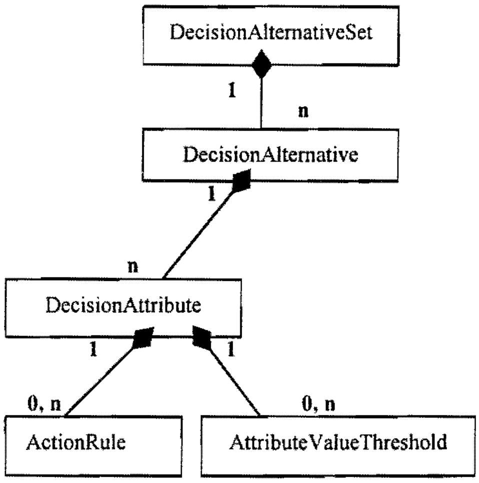
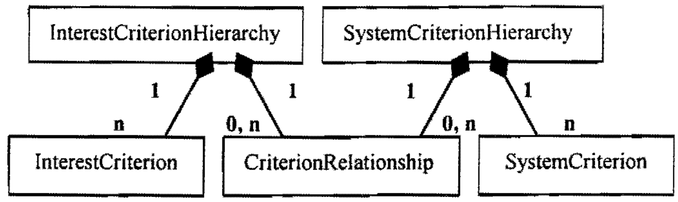


Figure D.9: Sub-diagram of Class Interactions for CaseDocument

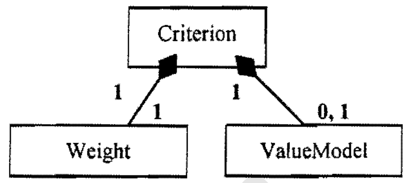
Appendix E: Class Aggregation Diagrams of the DSS Model



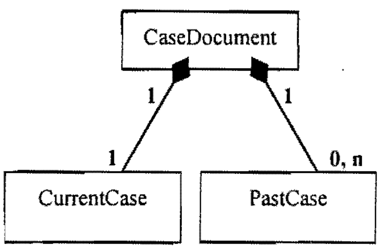
(a)



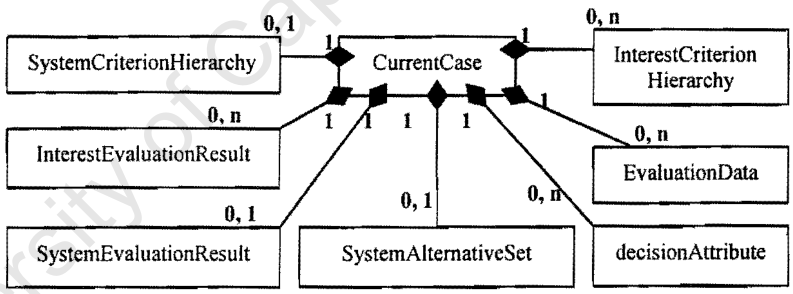
(b)



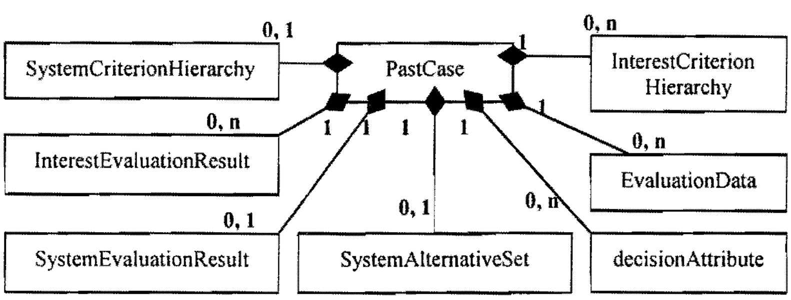
(c)



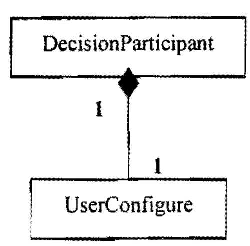
(d)



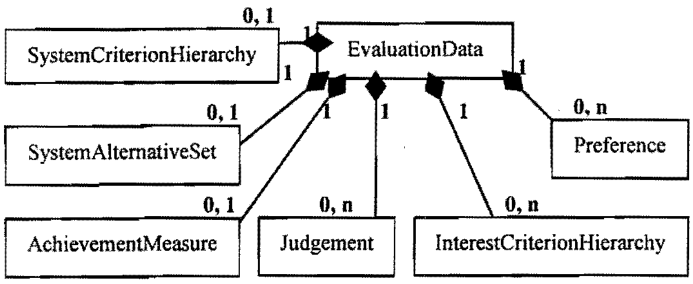
(e)



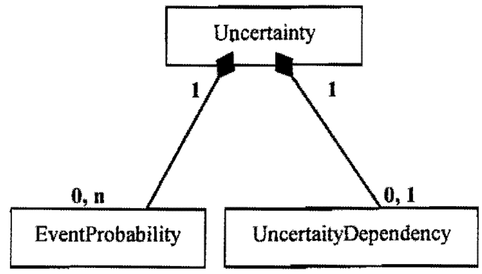
(f)



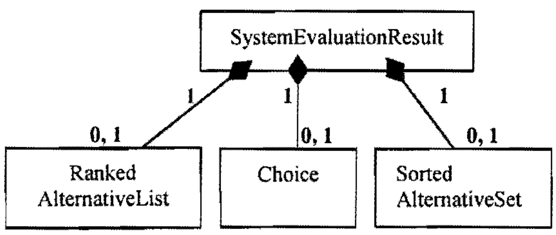
(g)


EvaluationData
SystemCriterionHierarchy
SystemAlternativeSet
AchievementMeasure
Judgement
InterestCriterionHierarchy
Preference
0, 1
1
0, 1
0, 1
0, n
0, n
0, n

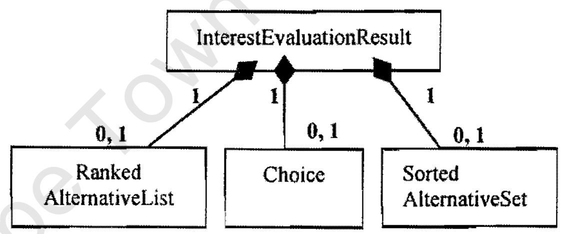
(h)


Uncertainty
EventProbability
UncertaityDependency
1
1
0, n
0, 1

(i)

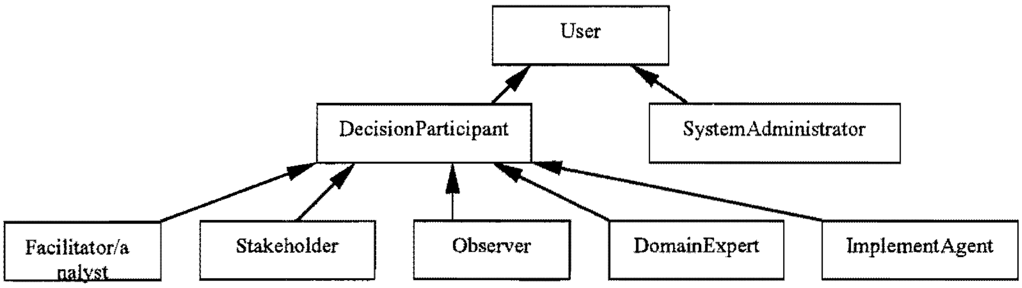

SystemEvaluationResult
Ranked AlternativeList
Choice
Sorted AlternativeSet
1
1
1
0, 1
0, 1
0, 1

(j)

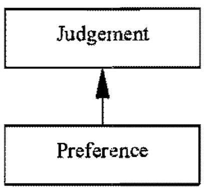

InterestEvaluationResult
Ranked AlternativeList
Choice
Sorted AlternativeSet
1
1
1
0, 1
0, 1
0, 1

(k)

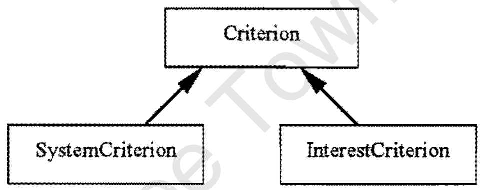
Appendix F: Class Inheritance Diagrams of the DSS Model



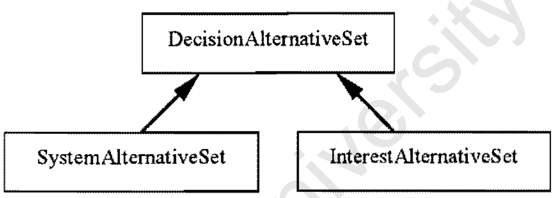
(a)



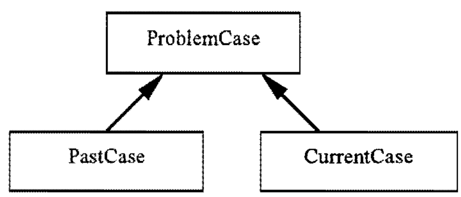
(b)



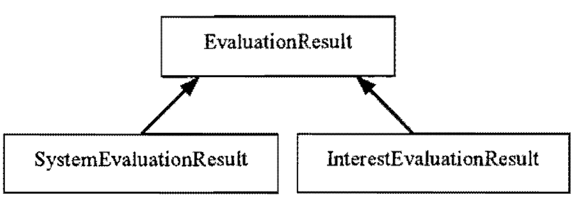
(c)



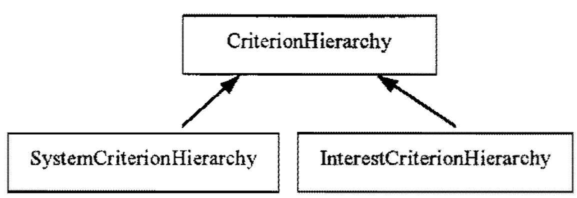
(d)



(e)

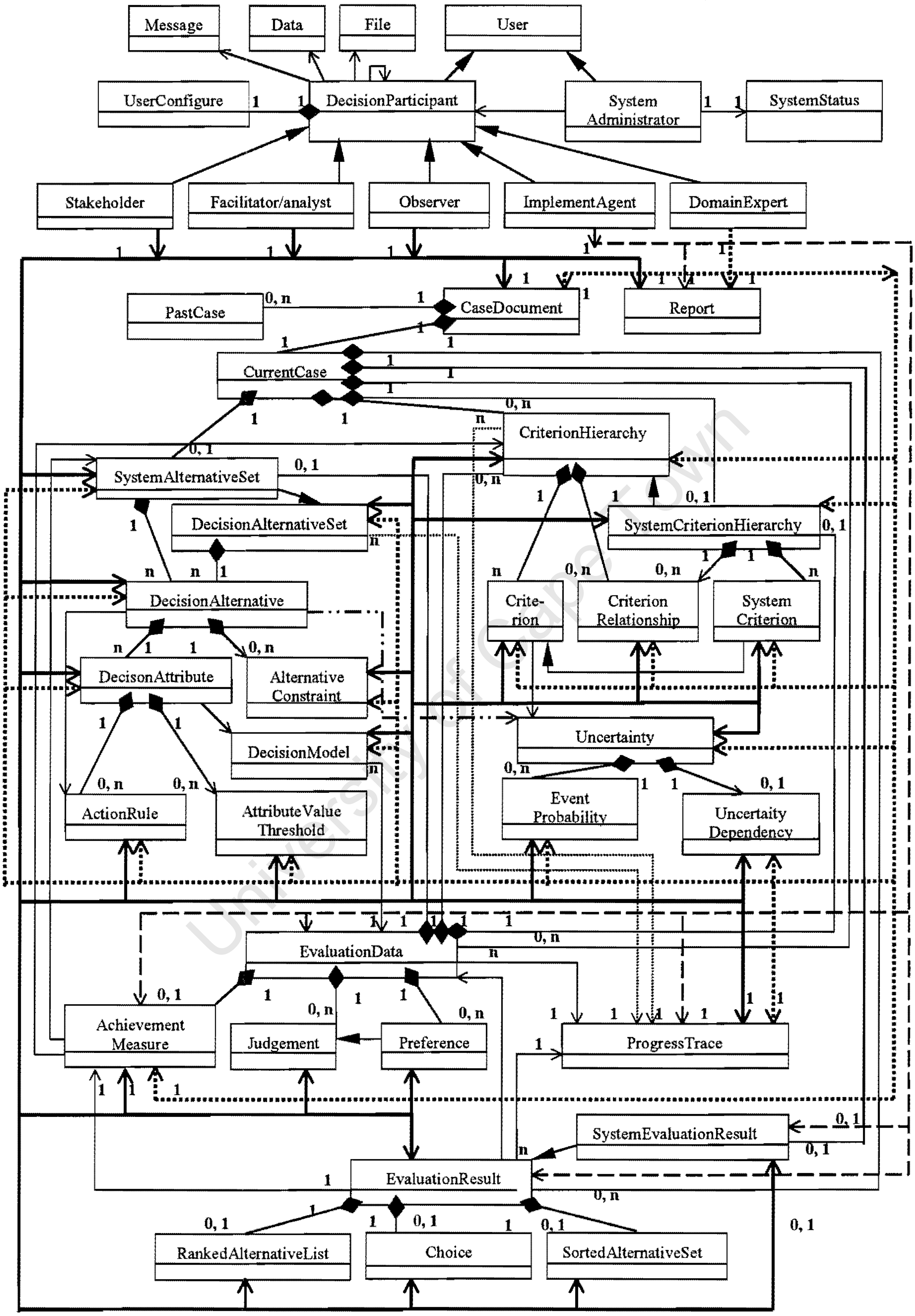


(f)



(g)

Appendix G: System Class Diagram of the DSS Model



Appendix H: The Development of WRC DSS

This appendix discusses the development of the Decision Support System for the Water Resources Commission (WRC DSS) by using the DSS model built in the study. Section H.1 introduces some aspects of the system, including system activities, user characteristics, general constraints, assumptions and dependencies. Section H.2 and H.3 discuss the system analysis and system design by utilising the DSS model respectively. Section H.4 describes the system implementation. Section H.5 contains conclusions.

H.1 System Description

The system is designed to be a web based MCDM group decision making support system for natural resources allocation. The MCDM method used in WRC DSS is a MAVT (Multi-Attribute Value Theory) based approach called Scenario-Based Policy Planning (SBPP) (Stewart, Scott and Joubert, 1993; Stewart and Scott, 1995; Stewart, Joubert, Scott and Low, 1996). The system will be able to do preliminary problem structuring and decision alternative evaluation at its minimum form. It will support all the phases of decision making at its full form. Group decision making is also needed in order to allow a group of decision makers working together as a team sharing information interactively, generating ideas and actions, choosing alternatives and negotiating solutions. This section describes the WRC DSS system in terms of general software specifications, including system activities, system users, and general constraints such as system assumptions and system dependencies.

(1) System Activities

The procedure of decision making in natural resource allocation has often been described as consisting of several distinct and iterative stages: problem structuring, evaluation, aggregation and implementation, as discussed in Chapter 2. WRC DSS will support all the stages except implementation, which mainly concerns the planning of tasks to carry out the decision made. In the stage of problem structuring, the problem is defined in terms of criteria, alternatives, and other related data that need better understanding for the problem under consideration. The evaluation phase elicits subjective judgements, value functions (a kind of decision model for evaluating alternatives), and value weights for measuring the trade-offs amongst objectives (criteria). The aggregation phase performs the calculation of the expected weighted value of each alternative. After all, the sensitivity of the weighted value to weights is examined. In short, problem structuring includes alternative generation, criterion identification, and criterion hierarchy construction. During the evaluation stage, alternatives generated are assessed according the criteria in the value tree. The aggregation stage focuses on the weighting of criteria and sensitivity analysis.

(2) System Users

Various participants in the water allocation decision making in South Africa can be identified after examining the DSS model. They include multiple stakeholders, domain experts, and other necessary facilitators/analysts. The users of the WRC DSS system include facilitators/analysts, stakeholders from local communities, research institutions, governmental organisations, and general users for educational purposes. These users have different cultural, economical and political backgrounds. They are dispersed geographically all over the country.

(3) General Constraints

The system must allow quick and easy retrieval or input of data, information, judgement and other knowledge through the interface. It must not be a burden for a user to use the system; a user should find that the product is a useful and worthwhile tool for decision making. Requirements to computer hardware, software, and network bandwidth are also considered. The system should be operable on average computers which have with access to Internet through ordinary ways. It is assumed that the computers used in the system for decision making have a colour monitor, network connection and a web-browser. The system runs under a web browser, such as Netscape Communicator and Internet Explorer (IE), through Internet although there are no specific requirements for operating platforms, other equipment, and utilities.

It is required that the system be available at all times although the availability of the system is dependent on the operation of the host computer system, which is known as the server in the system. The server is the information centre that keeps relevant data, and applications for the system and can communicate to the users dispersed geographically via the Internet.

The system description of general system specifications sets the context for the system development, which can be divided into three stages, i.e., system analysis, system design, and system implementation. The next section discusses the stage of system analysis.

H.2 System Analysis

System analysis deals with the basic requirements of the system, including primary system functions, system attributes, actors, use cases and classes. Actors, use cases, classes and class relationships are obtained based on the DSS model. The system is composed of classes, together with their properties, relationships and interactions. Static and dynamic system behaviour is illustrated by various diagrams representing classes and their interactions.

H.2.1 Brief Description of Functions

The system offers Internet base group decision making support. Group decision making can be carried out by several decision stakeholders, which represent different interests, and an analyst who acts as a facilitator, without geographical restriction (users need only Internet access) for a certain decision problem case, such as Sabie River water allocation. The main functions of the system are listed below according to different categories of users.

(1) Stakeholder (Each Interest Party)

A stakeholder is assumed to represent an interest party and have different points of view by which alternative policies can be judged or compared. The major decision making tasks carried out by a stakeholder are listed below including communications, criterion identification, alternative set examination and creation, criterion hierarchy tree construction, decision alternative evaluation, criterion comparison and analysis aggregation, and data saving.

(a) Communicate with each other via a notice board

Each user can make a short comment on the notice board on various subjects, such as the identification of criteria and alternatives, during different stages of decision making. Users can also browse the comments input by others.

(b) Identify (edit) criteria

For each specific interest corresponding to a stakeholder, criteria are identified to allow comparison of decision alternatives according to a particular point of view. These criteria can be modified at a later stage by the same stakeholder who identified them. Decision participants are also allowed to browse the criteria input by others.

(c) Examine decision alternative (scenario) sets

Users are allowed to view different decision alternative sets. Two kinds of alternative sets, i.e., foreground and background sets, are defined in the Scenario-Based Policy Planning (SBPP) approach (Stewart, Scott and Jourbert, 1993; Stewart and Scott, 1995; Stewart, Joubert, Scott and Low, 1996) for multicriteria decision making in natural resource allocation. Decision alternative is known as decision or policy scenario in SBPP to deal with broad complex policy issues that are possibly imprecisely defined as regards operational details. Foreground and background alternative sets are then called foreground and background scenario sets. A background scenario set is a manageable subset for interested parties to evaluate, i.e., a pool of scenarios that is sufficiently rich so that all interested parties can find a reasonably satisfactory alternative, perhaps by interpolation between scenarios. Through judicious interpolation, for instance, the principles of Experimental Design(although improved methods for this step are the subject of on-going research), virtually any scenario can be found within it. A foreground scenario set is needed for the participants to compare a few alternatives directly whose number of elements can not contain more than about 7 to 9 scenarios. Each user can view the background scenario set and the foreground scenario set for the problem case under consideration.

(d) Create personal scenario set

Extra scenarios can be added to the foreground set to create a personal scenario set by each decision participant possibly by adding small number of policy scenarios from the background set. Users, however, are not allowed to modify the existing scenarios in the foreground and background sets since these two sets are benchmarks for all decision participants. Modification to the foreground and background scenario sets is only through the facilitator/analyst.

(e) Construct value tree

The criteria identified for each specific interest corresponding to a certain stakeholder can be constructed into a criterion hierarchy tree by the stakeholder. A tree node represents a criterion, to be either composite or detailed. Node names can be selected from existing criteria input beforehand. New criteria can also be added by simply inputting a name for a node, and the upgraded criterion data is saved automatically when the tree is saved. Users are allowed to view the criterion hierarchy trees of others.

(f) Evaluation of scenarios

Policy scenarios are evaluated according to a specific point of view defined by a criterion in the criterion hierarchy tree. The evaluation is carried out directly on a thermometer scale or via graphs of value functions. Policy scenarios can be compared against each other on a thermometer scale from 0 (the worst) to 100 (the best). A graph can also be used to represent a value function by which measures of the relative degree to which policy scenarios satisfy the aspirations or desires indicated by a specific criterion can be automatically calculated. A value function is usually applicable when a criterion directly matches a scenario attribute, which one of the features or properties used to describe a decision scenario. Decision participants can view the evaluation data and the value functions by others. Different approaches are used to display the results of scenario evaluation. The evaluation results are reflected on a thermometer scale from 0 (the worst) to 100 (the best), and graphs are used to allow the display and modification of value function.

Bar charts are provided for the display of the element values corresponding to a certain scenario attribute for the scenarios in the foreground scenario set. This allows decision participants to examine the values of various scenarios before they compare these scenarios according to a certain criterion.

(g) Criterion weighting and aggregation

Criteria are weighted and the evaluation data is finally aggregated to produce analysis results. Child criteria of each node in the criterion hierarchy tree are compared against each other, being given different weights. These criteria are under the same parent node and at the same level of the criterion hierarchy tree. A thermometer scale from 0 (the least important) to 100 (the most important) is used for the criterion weighting. Users can also view the weight data by others. Once the weighting at one parent node is finished, the partial analysis result corresponding to this criterion hierarchy node, which represents a collective point of view of some scenario comparison principles, is then available for the decision participant to view. The accomplishment of the criterion weighting for the root node and all its child nodes of the criterion hierarchy tree indicates the availability of the analytical result for the corresponding stakeholder.

Basic sensitivity analysis is available over the sensitivity of the weight of a criterion to the satisfaction levels of policy scenarios. A thermometer scale dynamically shows the changes of the order of scenarios, resulted from the adjustment of the weight values of a certain criterion.

Several approaches are provided to check various outcomes, including criterion weights, and partial and overall analytical results. Besides a thermometer scale for displaying weighting results, bar charts offer another way to allow decision participants to check the values graphically. After the scenario evaluation and criterion weighting, partial and overall evaluation value path graphs, which show the weighted evaluation data for a set of criteria under a parent node in a criterion hierarchy tree, are available for each interest (not for the decision problem as a whole). Partial and overall ranked scenarios for each interest can also be shown after the evaluation and weighting.

(h) Store interest data

Every single element of information and data, including criteria, comments, policy scenarios, evaluation data, weights, and other relevant information can be stored and retrieved later on by users.

(2) Analyst/Facilitator

An analysts/facilitator is a special system user that facilitates the decision making activities at the system level. The main functions of an analysts/facilitator include the creation of the background and foreground policy scenario sets, system level criterion weighting, system overall analysis aggregation, and examination of the evaluation and weighting by stakeholders.

(a) Create (edit) the foreground set and the background set

The analyst/facilitator can create a background scenario set and a foreground scenario set, and distribute a copy to each other user in the system when they are completed or updated. The foreground scenario set is provided for stakeholders to generate their own personal scenarios sets while the background set is provided for the reference of stakeholders when generating their own foreground sets. The analyst/facilitator is allowed to view the personal scenario sets input by stakeholders to modify the existing foreground set.

A decision scenario is described with a set of decision attributes, which are the features or properties used to describe a decision alternative. The scale of measurement used to define a decision attribute may be cardinal (a direct numerical measure, which may be on a subjective scale), ordinal (a rank ordering of outcomes according to a particular feature), or nominal (a classification of outcomes into unordered classes). A textual mode of display is used for the value input of decision attributes.

(b) Overall weighting and aggregation

The analyst/facilitator can identify the system criteria, each of them corresponding to the interest of a stakeholder. A system criterion hierarchy tree is constructed and the corresponding user interests for the system criteria are specified. These criteria are weighted. Final overall analytical results are obtained for the decision problem as a whole if the evaluation and weighting data for each stakeholder is available. A final rank of policy scenarios for the decision problem case can be obtained. Like for each interest party, several approaches, such as thermometer scales, bar charts and evaluation value path graphs, are provided to demonstrate the outcomes. Basic sensitivity analysis at the system level can also be carried out in the same way as at each interest level.

(c) Data Browsing

The analyst/facilitator is allowed to examine various scenario evaluation and criterion weighting data by each interest party to ensure that each of the stakeholders uses the system in an appropriate way and that the input data is valid. The examination may also help the understanding of the value perceptions by the stakeholders and facilitate the decision analysis ultimately.

H.2.2 System Attributes

Besides its functions for direct decision making support, the WRC DSS system is designed to have non-functional attributes to offer quality decision support to system users. These attributes mainly include system extendibility, system security, and system robustness

(1) System Extendibility

The system can be easily extended to adapt to a different decision problem and new system functions. The system is designed for MCDM decision support for natural resource allocation with multiple decision participants. In an ideal form, it can support the decision analysis of a different decision problem with no revisions and even no modifications to the web pages displaying problem related information. In reality, minor modifications of the system, i.e., manual updating web pages for a new decision problem, are allowed to simplify the system development. System functional extension can be easily achieved under the philosophy of object orientation, as discussed in Chapter 1. Additional classes and class operations can provide extra functions for the system.

(2) System Security

The system is safe for communications and information sharing. The network technology will provide a secure way to access the system. Users are required to provide authorisations at the logging-in. Central system administration is available for system status monitoring.

(3) System Robustness

The software will manage the mal-operations of users and will continue to function correctly despite incorrect input. The tolerance to user errors in the system provides a robust way to ensure the stability of the system. Help or operation hints are available to the users in such an occasion of operation errors.

H.2.3 Actors and Use Cases

The system functional description and non-functional attributes determine the primary requirements for the system, which are then captured with use cases for each system actor. Actors are physical objects that have influence on the system under consideration when interacting with the system. A use case is a system function, a particular form or pattern of usage. A collection of use cases is like a checklist and describes the functions of a system. A detailed description, usually including a sequence of actions that illustrates a variation of a use case, is called a case description. The DSS model is utilised in the identification of these actors and use cases.

Actors in the DSS model include stakeholders, analysts/facilitators, domain experts, implementation agents, and observers. Some of these are not applicable in WRC DSS. For example, domain experts and implementation agents are not required in the system. These actors in the DSS model are considered irrelevant in the system. Observers in the model are translated into system learners in WRC DSS while other actors in the model are kept.

Use cases for the DSS model, which can be found in Chapter 5, are examined according to the system functions and non-functional attributes described in the last section. Use cases in the DSS model to display the final analytical results in three ways, i.e., a chosen decision alternative, ranked alternatives, and categories of sorted alternatives, are converted into one use case to display only one of them, i.e., ranked alternatives. Use cases in the DSS model to input decision models is detailed to allow graphical input of value functions for a corresponding scenario attribute to evaluate scenarios. Unnecessary use cases, such as criterion relationship expression, uncertainty consideration, past case demonstration, interfaces to external systems, and those use cases related to the deleted actors, are thrown away. The system actors and use cases defined resulting from the examination are listed below in the tables, which are similar to those in Chapter 5 for the summary of the system requirements of the

DSS model. Each table of Table H.1a-c represents a system actor and its use case set in WRC DSS.

Table H.1a: System Requirements (Facilitator/analyst)

Facilitator/analyst and its Use Cases	Brief Description
Configuration of user privileges	User privilege configuration
Send messages	Message broadcast and mail
Retrieve messages	Message reception and reading
Co-ordination of various decision making activities	Check and control of the progress of the decision making
Initialisation of relevant data	Initial data, information, and structure
Generation of alternatives for evaluation	Selection of an overall alternative set for evaluation by stakeholders
Construction of the system level criterion hierarchy	Construction of an overall criterion hierarchy
Evaluation of overall criteria and final aggregation	Comparison of the overall criteria and final aggregation of all scores
Sensitivity Analysis	Sensitivity analysis of aggregation results
Decision making guidance	Automatic information of decision making activities
Problem orientation	Textual and graphical problem document display

Table H.1b: System Requirements (Stakeholder)

Stakeholder and its Use Cases	Brief Description
Send messages	Message broadcast and mail
Retrieve messages	Message reception and reading
Brainstorming of the problem	Brainstorming problem perceptions
Construction of a criterion hierarchy	Construction of an individual criterion hierarchy
Generation/modification of alternatives	Generation/modification of individual alternatives
Evaluation of alternatives	Evaluation of the selected alternatives by the facilitator/analyst
Evaluation of criteria and individual aggregation	Comparison of the overall criteria and individual aggregation of scores
Sensitivity Analysis	Sensitivity analysis of aggregation results
Decision making guidance	Automatic information of decision making activities
Problem orientation	Textual and graphical problem document display

Case descriptions for these use cases are obtained easily by the examination of the case descriptions for the DSS model, which are shown in Appendix A, according to the actors and the use cases identified for WRC DSS. Some effort, however, is still needed to ensure that necessary translation and elaboration of the DSS model are made for some case descriptions, such as the one that deals with decision models, to fit into the specific system requirements. Once the use cases and case descriptions for each actor are available for the system, system classes can then be identified.

Table H.1c: System Requirements (Other System Actors)

Actor Use Cases	Brief Description
System Administrator	
Registration of users	User registration
Monitoring of system status	System status monitoring
Observer	
Trial use of the system as a stakeholder or a facilitator/analyst	Simulation of a stakeholder or a facilitator/analyst simulation
Send messages	Message broadcast and mail
Retrieve messages	Message reception and reading

H.2.5 System Classes

System classes are identified based on the DSS model by examining the use cases and case descriptions captured for the actors in WRC DSS. Classes are the basic elements for the building of a software system, and they define various decision problem entities and system components in a decision support system. The DSS model is examined, and those classes that are not contained explicitly or implicitly in the use cases and case descriptions of WRC DSS are removed. Class relationships, class interactions, and relevant diagrams are also updated to meet the requirements of the specific system. For instance, the unnecessary classes and the relationships are deleted from the class interaction diagrams and class diagrams in the model. In fact, classes and related DSS model elements provide a new point of view of the system structure for the first time during the system development procedure. It is obvious that the DSS model saves all the effort to carry out a detailed analysis to capture all these classes, class attributes, class operations, class relationships, etc.

A list of primary classes for WRC DSS, together with class attributes, operations, class relationships, and class interactions, are identified in this way. A system class diagram for WRC DSS is obtained by simply keeping the original structure of the class diagram for the DSS model while erasing those irrelevant classes and connecting lines. The attributes and operations of the classes remaining in the diagram are copied from the model. Classes for WRC DSS are shown in Appendix H-1 together with their definitions, attributes and operations. After the identification of the primary classes for WRC DSS, system design begins.

H.3 System Design

System design deals with how to organise the construction of a system. Logical and physical models of a system are constructed out of classes. Subsystems represent the logical structure of a system, while the system architecture indicates the physical structure as well as other physical deployment. The DSS model provides templates for both logical and physical structures for the development of the system.

H.3.1 Subsystems

The subsystems of WRC DSS are defined based on the DSS model. A subsystem is a logical collection of highly cohesive classes. A subsystem represents the physical arrangement of the software components based on a common property, for example, similar functionality and common physical location. The subsystems of WRC DSS are obtained by examining the subsystems in the DSS model and replacing the classes in the model with the identified classes of the system.

Figure H.1 shows the subsystems of WRC DSS. It basically has the same structure as that in the model. The problem understanding subsystem helps users understand the problem and the decision making with web pages and relevant documents. The subsystem of system administration administers the users and monitors the system. The problem structuring subsystem deals with identification of basic decision elements, such as alternative attributes and criteria, during the brainstorming phase, the generation of alternatives and a final alternative set for comparison of alternatives, and the construction of criterion hierarchies for criterion evaluation. The evaluation subsystem elicits stakeholders' preferences and finally aggregates to obtain analytical results. The user interface subsystem use windows, menus, tool bars, and dialogs, etc, to offer the users facilities to accomplish their tasks. The database subsystem manages various data to supply knowledge and information to decision participants and other subsystems.

The physical components of the system defined by the subsystem are deployed across some locations on hardware, as indicated in system architecture. The system architecture of WRC DSS is designed after the considerations of some technical aspects.

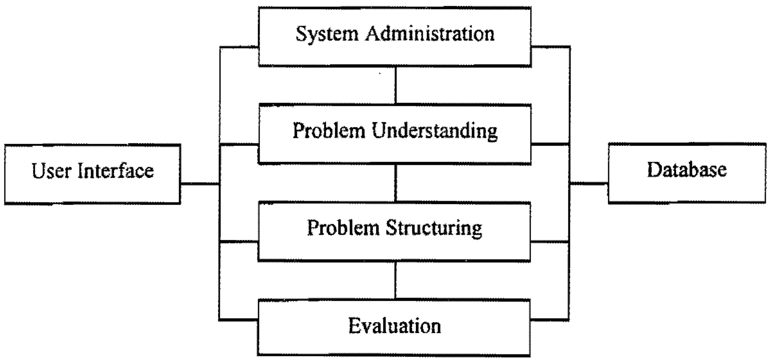


Figure H.1: WRC DSS Subsystems

H.3.2 Technical Considerations

Before the actual implementation of the system, tactical decisions, which concern common standards, practices and policies, are made in order to identify design requirements at the early stage. Some common design decisions involve the selection of an implementation language, persistent data storage, the look and feel of the user interface, error handling, and communication mechanisms.

The checklist in Chapter 8 provides a convenient way to examine the mapping of the DSS model and the system by ensuring appropriate implementation of the demanded requirements of the system. Some of the items in the checklist have already been considered in the previous stages of system development of WRC DSS. These items include value expression formats for some decision elements such as decision attributes, decision alternatives, decision models (value functions), and evaluation results, basic brainstorming support, MCDM method selection, criterion hierarchy construction, and evaluation representation, as described in Section H.2. Some of the items in the checklist are not applicable in the implementation of WRC DSS, and these items are ignored in the system. Table H.2 summaries the issues for technical considerations.

There are two broad categories of technical considerations when implementing WRC DSS. The first category is about general implementation considerations, which are primarily based on the DSS evaluation principles for technical implementation. The second one deals with decision making considerations. The general considerations for the system implementation include those of development environment, user interface, learning and use ease, help information, data management, system installation, and class library. The main technical approaches to deal with relevant issues are included in Table H.2.

Table H.2: WRC DSS Implementation Checklist

Item Name	Main Consideration Issues
General Implementation Consideration	
Development Environment	JBuilder; Java
User interface	Windows; Java style
Learning and Use Ease	Toolbar, Menus, keyboard commands, Orientation
Help Information	General on-line tutorial
Data Processing	MS SQL
System Installation	Java Applets
Class library	Private DSS class library; Java class libraries
Decision Making Consideration	
Problem Explanation	Web pages
Decision Elements	Spreadsheets for Scenario Set Generation

Java and JBuilder are used respectively as the programming languages used to code the system and the development platform in which the system is developed. Java is one of the main programming languages used for network solutions. It is operation platform independent. Java programs can run on almost any platform without modifying the codes. Java can be used in a variety of ways. Java Applet is one of the most publicised applications. Applets, which are used only on the client side of systems, are Java application components which are downloaded, on demand, to the part of the system which needs them. However, Java can also be used to create desktop applications and web servers and to extend web servers with customised processing. The latest Java technology provides Java Servlet techniques among other enhancements. Java Servlets are the Applet counterparts running on the web server side while Java Applets are Java programms running on the client's web browsers. Applets and Servlets may implement the same functionality. The difference among them is that Servlets do not have a user interface while Applets do since Servlets run inside servers and they do not need a graphical user interface. JBuilder offers a comprehensive set of visual development tools for building Java applications, Java-Applets, and Java Servlets, as well as JSP (Java Server Pages), Java Beans, Enterprise Java Beans (EJB) and distributed CORBA (Common Object Request Broker

Architecture) applications. Developers can deliver platform independent solutions, from applets to application, to support network computing.

The windows based user interface is designed with Java user interface components in the JBuilder development environment. Two Java class libraries, which are called Java AWT and Java Swing, offer various graphical tools and user interface items. They are used as user interface widgets to generate windows, menus, icons, dialogs, and other user interface elements for the user interface. Other issues of user interface, such as cursor and screen control, and tolerance to user errors and system responses are also considered.

The data base management system in WRC DSS is Microsoft SQL Server. SQL (Structured Query Language) is used as the database operation language. SQL Server is a relational database management system and provides a comprehensive platform that makes it easy to design, build, manage, and use network data warehousing solutions. As to data input and output in the WRC DSS system, web pages, dialog boxes, and spreadsheets are considered as basic formats. In addition, there is no constraints to the problem size the system is able to cope with.

Some class libraries are used in the development of WRC DSS. A preliminary private DSS class library developed for the study is used to support the building of decision support functions in the system. This DSS class library is very limited and can only support the design and Java implementation of some of the main classes of decision elements and other system components defined by the DSS model. Java class libraries, namely Java AWT, Java Swing, and Java Servlet, are used for the general system construction. Spreadsheet, trees, and thermometer scales classes are provided by Java Swing components to represent the alternatives, criterion hierarchy, and evaluation value respectively.

Other three general aspects of the system implementation are also considered. System operations are easily operable by users through toolbar, menus, and keyboard commands, and system orientation is also offered to ease the system learning and use. A general on-line tutorial and system status displaying are available to provide limited help information. System installation is achieved through the downloading of Java

Applets via the Internet, and there are no additional equipment requirements by the system.

The decision making considerations for the WRC DSS implementation mainly involve the representation of scenario sets and problem explanation. That is why the checklist defined in Table 8.1 is reduced to contain only two items after the system analysis. Other issues in the original checklist, including the implementation of major dynamic decision elements and some other decision making tasks, have been considered in the system analysis or are not applicable in the system. Various web pages, such as HTML (Hypertext Makeup Language) and JSP (Java Server Pages), are used for the purpose of problem explanation (and also for brainstorming, system status monitoring and identification of some decision elements). HTML (Hypertext Makeup Language) is the standard language for describing the contents and appearance of pages on the World Wide Web. JSP (Java Server Pages) is a script language that can present data provided by the server to the clients. JSP and Java Beans (reusable Java classes with specified formats) are used to dynamically generate web pages according to the data provided by the databases on the server. For the generation of decision scenario sets, spreadsheets are used as the basic form of input and output. Different scenario sets, including background, foreground, and individual scenario sets, are shown on spreadsheets, which can be created, modified, and saved by corresponding decision participants in different ways.

The decisions made towards the items of various technical considerations for WRC DSS have an impact on the detailed design and implementation of the system. Actual system programming and physical deployment of system components are considered on the grounds of the detailed implementation techniques. The next section discusses the design of the system architecture of WRC DSS.

H.3.3 System Architecture

The system architecture of WRC DSS is designed based on the DSS model. System architecture shows the physical computing units, the devices, and the communication-links between the computing units in a system. The devices support the execution of the computing units in a system. Every entity, including the computing units, the devices, and the communications, in the system architecture of the DSS model is examined for its location, functions, and relationships to other entities. An entity is thrown away if it is no longer necessary in a specific system. Some changes may also be undertaken to cater for the specific technical environment of a system. However, the main ideas and the overall structure may remain unchanged.

The system architecture for WRC DSS is shown in Figure H.2. It is almost a copy of that for the DSS model except there is only one server in WRC DSS while the architecture of the model can have an many as necessary. The Internet, the server, and front-end client machines are the three main kinds of devices in the system. The server is a computer that runs server packages to provide particular resources of Web pages and application processing services. The Internet is the largest WAN (Wide Area Network) in the world, consisting of a large collection of world-wide computers all networked together. Front-end client machines are normally local desktop machines, such as personal computers and workstations, which have access to Internet. Front-end client machines and the server are connected via the Internet.

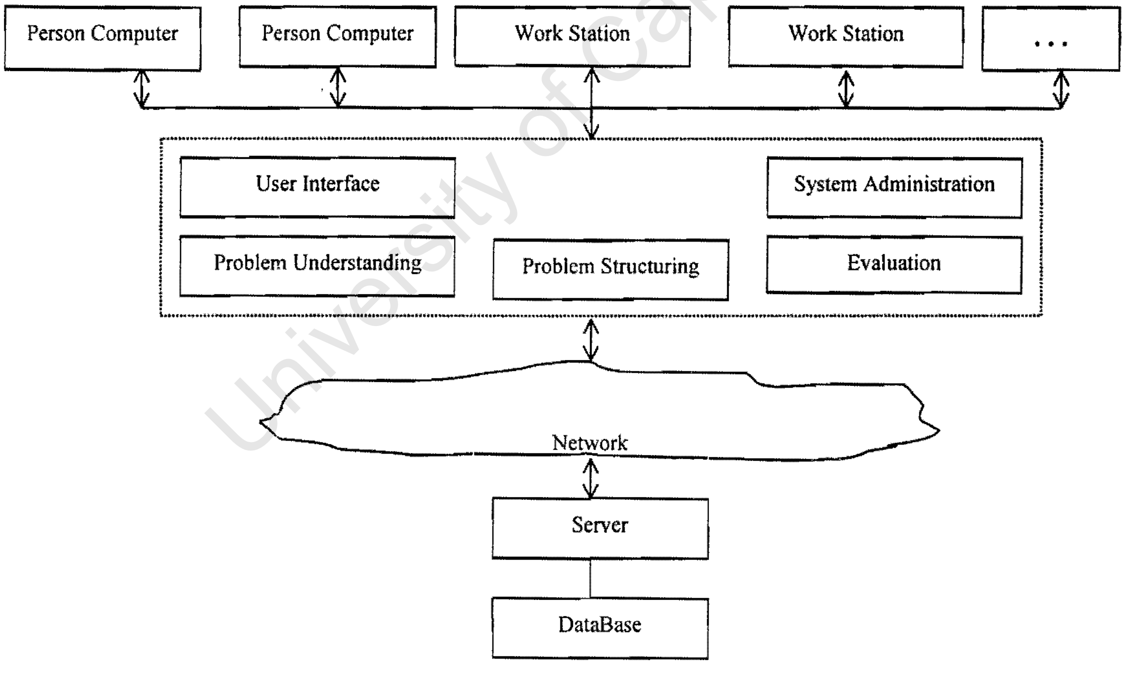


Figure H.2: WRC DSS Process Diagram

The computing units of the system are mainly located in the server. There are many computing utilities in the server, such as Web pages, application services, databases and their administration systems, which are called Data Base Management Systems (DBMS). Web pages provide some documents such as HTML (Hyper Text Marker Language) and JSP (Java Server Pages), graphics, data services upon request from a front-end machine running a Web browser. The code of the subsystems of system administration, problem understanding, problem structuring, and evaluation, like data in databases, reside in the server in the forms of Web pages, Java Applets, and Servlets. These subsystems are accessed through web browsers and operated via the user interface subsystem by using various client machines. The databases can be operated by users via Java Servlets. The network is the communication media that enables the execution of various applications and the retrieval of data and information without any constraints of geographical locations.

The underlying network computing technology for WRC DSS is the so-called client/server architecture. The object oriented Java programming and Internet browsers offer a wonderful opportunity for group decision support systems to be implemented in such a client/server mode. The WRC DSS system uses various Java techniques and the Internet to implement the client/server network computing architecture, which has three tiers, as shown in Figure H.3.

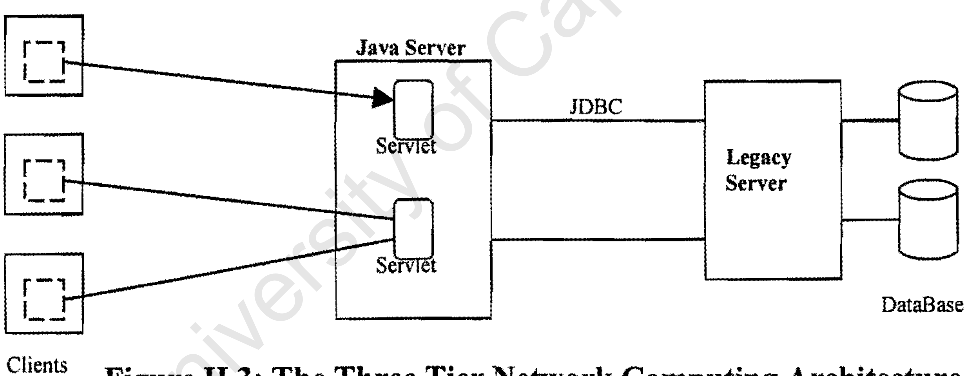


Figure H.3: The Three Tier Network Computing Architecture

Servlets play an important role in the client/server network computing architecture. The initial use of Servlets is to provide secure web-based access to data which is presented using HTML web pages, interactively viewing or modifying that data using dynamic web page generation techniques. For example, a Servlet might be responsible for taking data in an HTML (Hypertext Markup Language) order-entry form and applying the business logic to update a company's order database. Servlets have been developed to support full server services, and can be embedded in web servers. In addition, Servlets can talk to each other. Servlets frequently use some kind of persistent storage, such as files or a database. Static or persistent information can be shared across multiple invocations of the servlet, allowing information sharing between multiple users.

Java applets and Servlets are excellent tools to accomplish the tasks on the client and server machines. Java servlets run on the server side for data collection, data analysis and information distribution. Databases are stored on the server and can be accessible to the Servlets. This is a basic way that Servlets are used in the middle tiers of distributed application systems. Java Applets and other applications run on the client side to interact with the decision participants. Therefore, clients may range in complexity from simple HTML forms to sophisticated Java applets.

There are three tiers in the client/server network computing architecture used in WRC DSS. The first tier uses any number of Java enabled browsers, which are built on personal computers or workstations. Complex user interface tasks can be handled by Java applets downloaded from the second tier servers; simpler tasks could be handled using standard HTML forms. The second tier of the architecture consists of Servlets which encapsulate logic of the application. Servlets may be used to connect the second tier of an application to the first tier. The third tier of the system consists of data repositories.

H.4 Implementation

After the design of the system architecture, the implementation of WRC DSS is then started. The system implementation deals with the identification of classes, class operations, and class relationships, class relationship design, and class coding. Classes and class relationships identified during the system analysis phase are the primary input to the phase of implementation. Additional implementation-oriented classes are discovered out of considerations of some specific technical aspects, user interface, development environment, and the Java class library, etc. The additional classes identified for the implementation of WRC DSS at different time are listed below.

AggregationScaleSlider: A thermometer scale to indicate the ranking of alternatives

AlternativeEvaluationScaleSilder: A thermometer scale to indicate the preferences of alternatives according to a certain criterion

AlternativeEvaluationWindow: A window for evaluating alternatives and displaying the ranked alternatives

AlernativeSetSaveServlet: A Java Servlet to operate alternative set data
AlternativeTable: A spreadsheet for displaying and input the alternative data
AlternativeWindow: A window for constructing decision attributes, alternatives and alternative sets.
AttributeDialog: Application dialogue box to show and input attribute data
AttributeRangeSelection: A selection dialog box for displaying and inputting decision attribute thresholds
CriterionBean: Java bean to store and provide criterion data to and from a JSP page.
CriterionEvaluationWindow: A window for evaluating criteria and displaying the ranked alternatives
CriterionEvaluationScaleSilder: A thermometer scale to compare criteria
CriterionHierarchySaveServlet: A Java Servlet to operate criterion hierarchy data
CriterionHierarchyWindow: A window for constructing criterion hierarchies
CriterionJSP: Java JSP page to show and input criterion data
CriterionTree: A tree structure for displaying and input the criterion hierarchies
CurrentCaseBean: Java bean to store and provide problem case data to and from a JSP page.
CurrentCaseJSP: Java JSP page to show and input problem case data
CurrentCaseOrietationJSP: Java JSP page to show current problem case data and information
CurrentCaseServlet: Problem case database operating Java Servlet
EvaluationDataSaveServlet: A Java Servlet to operate evaluation data
EvaluationResultSaveServlet: A Java Servlet to operate evaluation result data
JudgementSaveServlet: A Java Servlet to operate judgement and preference data
MessageBean: Java bean to store and provide message data to and from a JSP page.
MessageJSP: Java JSP page to show and input message data
MessageSaveServlet: A Java Servlet to operate message data
ProgressTraceDialog: A dialogue box to show the progress status and the guidance
ProgressTraceSaveServlet: A Java Servlet to operate progress trace data
SystemStatusJSP: Java JSP page to show system status

UserBean: Java bean to store and provide user data to and from a JSP page.
UserConfigureBean: Java bean to store and provide user configuration data to and from a JSP page.
UserConfigureJSP: Java JSP page to show and input user configuration data
UserConfigureSaveServlet: A Java Servlet to operate user configuration data
UserJSP: Java JSP page to show and input user data
UserLoginServlet: A Java Servlet to manage the logging in of users
UserSaveServlet: User database operating Java Servlet
UserSession: Java session class to trace user logging time
ValueFunctionGraph: A graph to display and input the value function
ValueFunctionSaveServlet: A Java Servlet to operate value functions

The attributes, operations and relationships of the newly identified classes are also defined. The class attributes are determined by examining the technical considerations and the attribute of the classes in the DSS model, with which the newly identified classes associate. The actual implementation of some operation of existing classes is moved to some the new classes in order to make all the classes more cohesive and more balanced in taking responsibilities. For example, the "save()" operation serves only as a virtual operation (method in Java) with no implementation in the classes of "Criterion" and "SystemCriterion" while the operation is realised with different Java code in the corresponding Java Servlet classes. The relationship of class containment (one object of a class contains one or more objects of another class, for example, "userName" in the class of "MessageSaveServlet", and "decisionAlternativeSet" in the class of "AlternativeTable".) is implemented either by value (the actual contained class is declared in the containing class) or by reference (the actual contained class is not declared but only referred by a pointer, in the containing class). The relationship of "uses" is dealt with parameter passing in the implementation of operations (methods). Arrays and sets implement the multiplicity of relationships between classes. Other relationships such as inheritance are implemented with the Java language facilities.

The class hierarchy is also summarised to facilitate the actual coding tasks. The class "CreterionTree" is inherited from "DefaultTreeModel", and "AlternativeTable" from "AbstractTableModel" in the Java Swing package. All the Java Servlet classes are inherited from the class of "HttpServlet" in the Java Servlet package. The window classes in the system are subclasses of "JFrame", the dialog classes are of "JDialog", and the scale slider class of "JSlider" in the Java Swing package. These are the major relationships of inheritance between the classes from the Java class libraries and the design classes of WRC DSS. Ideally, the classes for decision elements and some other DSS system components modelled in the DSS model should be created as child classes of those in the DSS model to cater for specific requirements of individual decision support systems.

After the identification of additional classes, class operations and class relationships for system implementation, the actual coding of classes starts. All the classes are implemented with Java. The coding effort for some classes is saved since the private DSS class library offers their existing implementation that can be used directly in the system programming.

The completed system is composed of a system orientation component and six subsystems. Some of them, such as the system orientation, only contain web pages while most of them consist of some clusters of classes. System orientation is to help users get familiar with the system and help them with the operations of the system. The six subsystems are system administration, problem orientation (understanding), problem structuring, evaluation, user interface, and database. System administration is responsible for the administration of users such as stakeholders, facilitators and system administrators. It also registers problem cases for different teams of users so that users in the same case may work together for the same problem case. System administration can show which user has accessed the system and when. This might be useful to co-ordinate different users. The subsystem of problem orientation helps understand the problem case. The subsystem of problem structuring deals with problem structuring activities for users to identify (edit) criteria, send and receive messages, construct criterion value tree, and view scenarios (facilitator can build the scenario set while normal decision participants may add extra scenarios). The subsystem of evaluation allows users to compare scenarios according to the criteria and to weight criteria and aggregation. The subsystem of user interface contains various interface facilities such as windows, menus, dialogs, etc. The subsystem of database enables data to be stored in the server and manipulated via operation Servlets.

H.5 Conclusions

The WRC decision support system (WRC DSS) is a system developed for water resources allocations in South Africa by using the existing DSS model. The development of the WRC DSS system demonstrates how the DSS model and the technical checklist designed in the study can be used. The DSS model offers an efficient and effective way to develop individual decision support systems.

The system has about 20,000 source lines of code, and was developed with an effort of about four person-months. A person-month represents the work of one average developer in one month's time. The effort is relatively little for a system of this scale (Albrecht and Gaffney, 1983; Bassman, McGarry, and Pajerski, 1995). Efficiency is obtained for the system development. A slightly revised version of the system is used for evaluation in the workshops in Chapter 8. It was shown that the demanded system requirements are met and the required DSS performance evaluation principles are satisfied by the system. Refer to Chapter 8 for further information.

Appendix H-1: Definitions, Attributes and Operations of Primary Classes of WRC DSS

AchievementMeasure: An algorithm to measure the relative degrees to which alternatives satisfy the aspirations or desires of a stakeholder, representing his/her overall strengths of preference between outcomes.

Attributes: name, stakeholderName, alternative, value, algorithm, description

Operations: input(), modify(), check(), calculate(), isComplete()

AttributeValueThreshold: Value ranges which action elements fall in.

Attributes: name, attributeName, valueProperty, valueRange

Operations: identify(), modify(), delete(), check(), examine(), save()

Criterion: a principle allowing comparison of decision alternatives. It is a tool to compare alternatives according to a particular significance point of view.

Attributes: name, inputUser, inputTime, parentCriterion, childCriteria, description

Operations: identify(), modify(), delete(), check(), save()

CriterionHierarchy: The hierarchy of relevant criteria.

Attributes: name, userName, criterionTree, criterionRelationships

Operations: create(), modify(), check(), isComplete(), evaluate(), save()

CurrentCase: The problem facing the stakeholders and its related data. It may include the relevant information to describe the problem case.

Attributes: name, information, alternativeAttributes, decisionParticipants, systemAlternativeSet, criterionHierarchy, systemCriterionHierarchy, evaluationData, evaluationResult, systemEvaluationResult

Operations: register(), modify(), delete(), check(), orient()

Data: Data that keep the information for the system, such as alternative data and evaluation data. It may be exported to and imported from external systems.

Attributes: name, format, type, values

Operations: copy(), paste()

DecisionAlternative: A description of one possible plan of action for the future.

Attributes: name, decisionAttributes, attributeValues, parentAlternative, childAlternatives, alternativeConstraints

Operations: identify(), modify(), check(), delete(), save()

DecisionAlternativeSet: A set of alternatives for individual users.

Attributes: name, count, alternatives

Operations: create(), modify(), check(), isComplete(), add(), delete(), save()

DecisionAttribute: The features or properties used to describe a decision alternative. It might be cardinal (a direct numerical measure), ordinal (a rank ordering), or nominal (unordered classes).

Attributes: name, valueProperty, valueThreshold, actionRules

Operations: identify(), modify(), delete(), check(), save()

DecisionParticipant: A virtual parent class for the classes that participate the decision making.

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note, problemCase, userConfigure

Operations: login(), register(), modify(), delete(), check(), checkConfigure(), modifyConfigure()

EvaluationData: The evaluation data of alternatives and their corresponding criteria at different stages.

Attributes: name, stakeholderName, alternativeSet, criterionHierarchy, judgements, preferences, achievementMeasure

Operations: check(), calculate(), input(), save()

EvaluationResult: The evaluation outcomes of various stakeholders and facilitator/analysts. It might in one of the three forms: choice, ranked alternatives, and sorted alternatives.

Attributes: name, stakeholderName, choice, rankedAlternativeList, sortedAlternativeSet

Operations: aggregate(), check(), save()

Facilitator/analyst: A person who facilitates the decision making processes.

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note, problemCase, userConfigure, isImitate

Operations: login(), register(), modify(), delete(), check(), checkConfigure(), modifyConfigure(), imitate()

InterestAlternativeSet: A set of alternatives generated by individual stakeholders.

Attributes: name, count, alternatives

Operations: create(), modify(), check(), isComplete(), add(), delete(), save()

InterestCriterion: Criteria of the individual stakeholder's level, which usually reflects different concerns of individuals or groups.

Attributes: name, inputUser, inputTime, parentCriterion, childCriteria, description

Operations: identify(), modify(), check(), delete(), save()

InterestCriterionHierarchy: The criterion hierarchy of each interest group

Attributes: name, userName, criterionHierarchy, criterionRelationships

Operations: create(), modify(), check(), isComplete(), evaluate(), save()

InterestEvaluationResult: The evaluation outcomes for individual stakeholder.

Attributes: name, stakeholderName, choice, rankedAlternativeList, sortedAlternativeSet

Operations: aggregate(), check(), save()

Judgement: Stakeholder's general judgement on alternatives or criteria.

Attributes: name, judge, judgeItem, referItem, valueProperty, valueRange, value, description

Operations: elicit(), modify(), check(), save()

Message: Message sent and retrieved by individual system users.

Attributes: name, priority, content, sender, receiver, sentTime

Operations: send(), broadcast(), retrieve()

Observer: A person who wants to study the system for various reasons.

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note, problemCase, userConfigure

Operations: login(), register(), modify(), delete(), check(), checkConfigure(), modifyConfigure()

Preference: Stakeholder's preferences to the alternatives according a certain criterion.

Attributes: name, judge, judgeItem(alternative), referItem(criterion), valueProperty (preferenceProperty), valueRange(preferenceRange), value, description

Operations: elicit(), modify(), check(), save()

ProblemCase: A decision problem in natural resource management domain. It may include the relevant information to describe a problem case.

Attributes: name, information, alternativeAttributes, decisionParticipants, systemAlternativeSet, criterionHierarchy, systemCriterionHierarchy, evaluationData, evaluationResult, systemEvaluationResult

Operations: register(), modify(), delete(), check(), orient()

ProgressTrace: A library to keep the knowledge of the decision making procedure and the status of the current decision making. It also contains the stages into which the whole decision making procedure is divided, and the instructions given to users as to how to do next in the decision making processes.

Attributes: decisionStages, decisionParticipants, decisionInstructions, decisionStatuses

Operations: monitor(), instruct(), update()

RankedAlternativeList: A ranked list of alternatives according to a set of criteria.

Attributes: name, stakeholderName, rankedAlternatives

Operations: check(), save()

Stakeholder: A real world person who is able to express preferences of alternatives and is usually a stakeholder as well.

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note, problemCase, userConfigure, isImitate

Operations: login(), register(), modify(), delete(), check(), checkConfigure(), modifyConfigure(), imitate()

SystemAdministrator: A person who administrates the system

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note

Operations: login(), register(), modify(), delete(), check()

SystemAlternativeSet: An overall representative set of alternatives of the system. The final version of this set is used by all users to evaluate the alternatives.

Attributes: name, count, alternatives

Operations: create(), modify(), evaluate(), check(), isComplete(), add(), delete(), save()

SystemCriterion: Criteria of the system level, which usually reflects different interests of individuals and groups.

Attributes: name, inputUser, inputTime, parentCriterion, childCriteria, description

Operations: identify(), modify(), delete(), check(), save()

SystemCriterionHierarchy: The system hierarchy of relevant criteria.

Attributes: name, userName, criterionTree, criterionRelationships

Operations: create(), modify(), check(), isComplete(), evaluate(), save()

SystemEvaluationResult: The overall evaluation outcomes for the system.

Attributes: name, stakeholderName, choice, rankedAlternativeList, sortedAlternativeSet

Operations: aggregate(), check(), save()

SystemStatus: The running status of the system.

Attributes: userStatus, resourceStatus

Operations: monitor()

User: A virtual parent class for all system user classess.

Attributes: name, passWord, organization, phoneNumber, emailAddress, postAddress, physicalAddress, webSite, FTPsite, otherAddresses, note

Operations: login(), register(), modify(), delete(), check()

UserConfigure: The configuration of access and communication priorities for users such as stakeholders, and facilitator/analysts, etc.

Attributes: accessPriority, communicationPriority

Operations: setAccessPriority(), setCommunicationPriority(), getAccessPriority(), getCommunicationPriority()

ValueFunction: Automated functions whereby alternative attributes can be analysed in response to the evaluation of alternatives.

Attributes: name, inputData, outputData, algorithm

Operations: identify(), modify(), calculate(), check(), delete(), save()

Weight: An indicator of importance of a criterion in comparison with others in the same level of a criterion hierarchy.

Attributes: name, criterionName, criterionHierarchyName

Operations: input(), modify(), check(), save()

Appendix I: The Decision Problem of the Deforestation of Table Mountain

1. The Background of the Decision Problem

Table Mountain, South Africa, is situated at the northern end of the Cape of Good Hope Peninsula. It encompasses Lion's Head and Signal Hill on the northwestern side, the centrally situated Upper Plateau which is bisected by Platteklip Gorge, and extends down in a southerly direction to form the Back Table (or Lower Plateau). The broad valley of Orangekloof is situated in the southern part, flanked by Constantia Corner and the Twelve Apostles.

Table Mountain has been described as a unique natural wonder, owing to the rich and diverse flora growing in a setting of majestic proportions and extraordinary beauty. Today there is an extremely high concentration of threatened plants on the Cape Peninsula, many of which grow on Table Mountain. Table Mountain has exceptionally high levels of biodiversity. It has approximately 1 470 species of plants – more than the entire British Isles, many of which are endemic. Table Mountain's natural vegetation, fynbos (literally, "fine bush"), is an evergreen vegetation characterised by hard leaves, thriving on nutrient-poor soil, and able to survive the Cape's wet winters, destructive winds and hot summers. The most well-known groups are Proteas, Erica and Restios (reeds), but other groups include a variety of daisies, legumes and buchus.

Table Mountain also provides countless recreational and tourism opportunities which are considered to be potential sources of income and job creation in the region. As a result of this, the greatest threat to the integrity of Table Mountain will come from development in response to the demands of tourism and recreation. Decisions around the management of recreational use and tourism will potentially impinge on the diversity and the scenic beauty of the landscape but will also have implications for the economic upliftment of the area.

During the 300-odd years since colonisation, the mountain has been exploited, abused, burnt, eroded, vandalised and despoiled. Within a few decades of the establishment of the first settlement at the Cape in 1652, the mountain was stripped of readily available timber for shipbuilding, housing, furniture-making, charcoal-making and lime kilns. With the growth of the settlement the demand for wood increased and the wood collectors moved higher up the mountain. As trees are slow growers, the wholesale collection of timber and firewood had a marked effect on the appearance of the mountain. Eventually it became necessary to introduce trees to green these slopes, and to assist in controlling soil erosion. In fact there was such strong public reaction to the bare and desolate appearance of the mountain, that in the 1880s an afforestation programme was launched to "clothe these barren slopes", and also to provide much-needed timber. The afforestation was so successful that the barren slopes were soon overgrown with cluster pine (Pinus pinaster), which spread rapidly to the detriment of the indigenous flora, also causing a public outcry. The plantations are of importance in terms of aesthetic and shade values, but are damaging to the natural ecosystem. Alien trees and shrubs are the major threat to the species-rich fynbos ecosystems of Table Mountain. If unchecked, aliens such as pine trees, Port Jacksons, Hakea and Rooikrans could destroy the Mountain's natural plants within 70 years (Hey, 1996).

The afforestation has links to other factors which detrimentally affect the ecosystem of Table Mountain. For instance, accidental fires have devastated indigenous vegetation, causing soil erosion and mudslides. Alien vegetation has smothered the Mountain's delicate fynbos. The rapid spread of alien vegetation on the mountain, the gradual destruction of the natural plant cover, and the resultant soil erosion caused by frequent fire and overexploitation have been a matter of increasing concern among citizens.

Specifically, the fires of recent years in the mountains around the South Peninsula and recently on Table Mountain have prompted action to rid Table Mountain of invasive alien vegetation. The devastating fires have highlighted the role of alien vegetation in spreading fires. Fynbos is highly inflammable and is a fire-adapted system needing fire to regenerate. Fynbos plants create relatively little fuel and there is not much combustible vegetation litter on the ground. This means the fire burns through an area quickly and does not create much heat or damage the soil. By contrast, alien invasive plants create much bigger fuel loads, which lead to hotter fires which can cause substantial damage to the soil by altering its chemical structure to make it less able to absorb water (hence promoting floods and mudslides) and destroying the seedbank still in the soil. Although fynbos is well adapted to fires, if veld is burnt too frequently those plants that require years to mature and produce seed will be gradually eliminated. Such fires also result in a heavy mortality of smaller creatures, particularly reptiles and nesting birds which are unable to escape the flames, and the loss of their food supplies in the form of fruits, seeds and insects.

The problem of the eradication of alien plants has arisen due to the damage caused to many aspects of Table Mountain. While the rights of the individual landowners and communities are recognised and must be honoured, no development which could adversely affect natural systems, the habitat of rare fauna and flora, scenic or historic features, or could result in the general degradation of the environment can be supported. Any development within the nature area must blend with the environment and be aesthetically acceptable. The Hey commission appointed in 1977 to investigate and report on the future management and control of the mountains of the southern Peninsula, urged that the conservation of flora and fauna receive prime consideration, that public recreation be secondary, and that no activity or works to the detriment of the natural environment be permitted. It was recommended that the management policy be based on a combined programme of nature conservation and outdoor recreation, embodying the principle of multiple usage. The general strategy of alien clearance suggested by Hey (1996) is to clear the sparsely infested areas of aliens and to contain the very dense stands until they can be removed.

The issue of deforestation of Table Mountain is in fact about the clearance of aliens. There would have to be an acceptable balance between social, economic, and conservation goals in the decision making process of the deforestation. Rights of various stakeholders, including individual landowners and communities, must be respected, while considering the conservation of the environment. That is also what brings this hypothetical decision support workshop into play.

2. The Decision Support System

A computerised system (called decision support system) is to be used to support the decision making procedure for the decision problem of the deforestation of Table Mountain. Users can access the system through Internet at their own computers according to their own time schedule. An orientation of the system is available through the Internet.

In our workshop, users, each acting in a role of a stakeholder in the decision problem in order to participate the decision making, are asked to sit together in the computer lab of the Statistical Department at UCT (University of Cape Town). Assistance to use the system can be offered and the whole procedure is to be carried out in a scheduled period of time.

2.1 Objective to achieve as a result of the decision making procedure and the decision support system usage:

What is the problem
What are the concerns
How to structure the concerns
What are the decision option elements and decision options
Evaluation of options
A final choice

2.2 The Generalised Knowledge of Land Use Decision Analysis

The existing generic knowledge for certain aspects of the problem is generalised from experiences and literature and is provided in the system for users' support. However, people should not be limited by the provided knowledge and are encouraged to think creatively by asking what other considerations should be taken all the time. The structures of main categories of existing knowledge are listed below.

3. The Decision Making Procedure

The decision making procedure is artificially divided into four phases.

3.1 First Phase - Initial understanding

The objective of this phase is to achieve initial understanding of the decision problem as a result of the analysis of the problem context, the people involved, and other elements.

3.1.1 The Problem Context

The problem context of the deforestation of Table Mountain is a kind of land use (afforestation/deforestation) in South Africa. There are very few natural forests, and forests of alien trees, which are planted out of various reasons (such as the demand for timber and wood pulp, and historic reasons), bring about dangers to naturally indigenous flora and fauna, which is threatened throughout Africa.

Factors that influence the decision problem are roughly classified into four categories: social, economic, political, and environmental factors. For each of the influence factors, there are still more subcategories of factors to be considered. Some examples of factors are listed below:

Political, e.g., political party policies
Economic: e.g., tourism, timber, industry
Social: e.g., fires, employment, housing, aesthetic, personal well-being
Environmental: e.g., conservation (flora, fauna); soil erosion, water catchment

Please think about the problem context and try to identify all possible factors involved.

3.1.2 The People and Stakeholders

Some people may have influence on the decision problem, for example, a physical person or group of persons who hold the same stake. They are usually needed to be discovered in the initial stage of decision analysis. These people mainly fall into the following five categories:

Local Communities (who affect or are affected directly or indirectly by the decision, for example, the physical district where a forest is located, and a commercial company that uses the forest as raw material)

Neighbour Communities (who affect or are affected directly or indirectly by the decision and who are geographically apart from the site where the decision problem is, for example, a neighboring district)

Governmental Organisations (National or district organisations who hold the policy and implement the final choice of the decision analysis, for example, the provincial water allocation committee)

Non-governmental Organisations or Societies (who hold the same interest related to the decision to be made, for example, a mountaining club)

Domain Experts (who offer domain knowledge in various aspects)

Stakeholders are among the people discussed above, e.g., local or neighbour communities, organisations, or their representative individuals. A stakeholder is a real world person who usually has a stake in the decision problem (and is expected to express preferences of different decision options at a later stage).

Please ask yourself: who are the people involved in the decision problem under each category, and who are the stakeholders.

3.1.3 Criteria, Option Elements and Decision Options

A criterion is a concern according to a particular significance point of view, allowing comparison of decision options. A criterion may include several sub-criteria. Some examples of major criteria include:

Personal well-being
Aesthetics/greenspace
Fires
Recreation
Tourism
Industries
Water effect
Conservation
Soil erosion

Relevant criteria are then organised to allow systematic comparison of decision options as well as criteria. There are different levels of criteria. Criteria of the overall level summarises different interests of all stakeholders while criteria of the individual stakeholder reflects concerns of individuals or groups. A **TEMPLATE** to organise all relevant criteria of all stakeholders is shown below:

Social:
- Personal well being
- Fires
- Recreation
- Aesthetics/greenspace
- Others

Economic:
- Tourism
- Other industry
- Sustainability of economic growth

Environment:
- Conservation
  - Species richness
- Soil erosion
- Others

Water
- Water catchment
- Others

All the criteria related to a specific stakeholder are organised into one structure together to indicate the point of view of the stakeholder. The criterion structures of all stakeholders are then organised to represent the overall concern or objective of these stakeholders.

Option elements indicate aspects of the action whose collective set constitutes the attributes for a decision option. There appear to be three main categories of option elements:

Levels of expansion (shrinkage): Should the deforestation take place in the entire region or only part (if so, how much) of it

Possible zones: Which zones should be considered for deforestation? On which basis these selections are based, for example on conservation, aesthetic or recreational issues?

Time span: The time period of deforestation/afforestation, for example, 5 years, 20 years or 50 years

Users are asked to act in a role of a stakeholder. They identify some decision options out of option elements and their own concerns.

Please try to identify all possible criteria (you may refer back to the problem context and take those factors into consideration as well), all possible option elements (not limited to the three categories), and the decision options you would like to bring up.

3.2 Second Phase – Further Understanding

The objective of this phase is to further understand the decision problem.

Users, who act as a stakeholder, associate and organise the relevant concepts discussed before in a diagram, and try to figure out their behaviours related to the deforestation of Table Mountain. They examine what their main concerns are, what their objectives are, what are the option aspects (action element) that they can use to deal with their concerns and to achieve their objectives, and what their possible options are.

After this analysis, please identify all possible concerns (criteria), option elements, and decision options.

3.3 Third Phase - Structuring

In this phase, decision criteria are organised into tree structures (with levels of hierarchies), and decision alternatives are finally generated.

A user organises all the identified criteria (which are related a specific stakeholder) into a tree structure with possible reference to the template of criterion organisation. Out of the decision options proposed by different users, a final set of decision options is generated by the facilitator of the workshop for their comparison.

3.4 Fourth Phase – Evaluation and Choice

In this phase, users are asked to express their preferences of decision options according to a specific criterion in the constructed criterion trees, and also to indicate the relative importance of criteria. At a final stage, a choice can be then made.

An analysis can also be conducted to see the sensitivity of the choice to the changes of relative importance (also called weights) of some criteria. This analysis is called sensitivity analysis.

A questionnaire is prepared for you to assess the performances of the decision making methodology and the system you have used. Please fill the questionnaire and send it back to the facilitator of the workshop.

Questionnaire for the Evaluation of the Performances of the methodology and the System

The questionnaire below examines your personal perceptions of a variety of aspects of the decision analysis methodology and the decision support system you have used. Please rate the methodology and the system's performance in each of the items in table below by circling the appropriate number on the scales provided.

Very Poor		Poor		Moderate		Good		Excellent
1	2		3	4	5		6	7

Table N.1: Questionnaire for the Evaluation of the Methodology and the DSS

Evaluation Item	Score						
1. Overall Confidence of the Solution	1	2	3	4	5	6	7
2. Methodology Performance							
Value of the generalized knowledge provided	1	2	3	4	5	6	7
Quick understanding of the problem	1	2	3	4	5	6	7
Ease of problem analysis	1	2	3	4	5	6	7
Ease of problem formulation	1	2	3	4	5	6	7
3. System Functional Performance							
3.1 Support of group decision making	1	2	3	4	5	6	7
3.2 Guidance in the decision making processes	1	2	3	4	5	6	7
3.3 Problem Analysis and structuring							
3.3.1 Support of problem understanding	1	2	3	4	5	6	7
3.3.2 Support of brainstorming	1	2	3	4	5	6	7
3.3.3 Support of problem structuring	1	2	3	4	5	6	7
3.4 Evaluation and Choice							
3.4.1 Support of judgement elicitation	1	2	3	4	5	6	7
3.4.2 Support of sensitivity analysis	1	2	3	4	5	6	7
3.4.3 Display of the results	1	2	3	4	5	6	7